'68' MICRO JOURNAL

$2.50 USA

Australia	A $ 4.00	New Zealand	NZ $ 4.00
Singapore	S $ 8.00	Hong Kong	H $20.00
Malaysia	M $ 8.00		

VOLUME III ISSUE XII • Devoted to the 68XX User • December 1981

"Small Computers Doing Big Things"

PHOTO CREDIT - NASA

YOUR CHOICE-smart either way

- Over 140 software driven functions
- 82 x 24 or 82 x 20 screen format — software selectable
- High resolution 7 x 12 matrix characters — P-31 green phosphor
- Upper/lower case character set — plus graphics character set
- 56-key alphanumeric keyboard — plus 12-key cursor, numeric pad
- Internal editing functions — insert, delete, scroll, roll, slide, etc.
- Parallel printer I/O port
- 50 to 38,400 baud operation — programmable
- Cursor type, cursor position, print control characters, protected fields, shift inversion, dual intensity and many other features

8212 — twelve-inch diagnonal screen or 8209 — nine-inch diagnonal screen

SWTPC

SOUTHWEST TECHNICAL PRODUCTS CORPORATION
219 W. RHAPSODY
SAN ANTONIO, TEXAS 78216 (512) 344-0241

Pascal for 6809

Pascal for the 6809 is a true native code compiler. Unlike the usual P-code Pascals which run in an interpretive manner, ours produces efficient assembly language mnemonics which can be assembled and run directly. This compiler is available for both 6809 FLEX™ and UniFLEX™. Many features not found in other Pascal systems were implemented while avoiding those features completely non-standard. Features of the Pascal system include:

- Supports most of Jensen and Wirth specification
- Produces fast and efficient 6809, native code
- FLEX run-time package may be trimmed
- Double precision real numbers (16.8 digits)
- Implements scalar, subrange and structured data types
- Standard I/0 using file buffer pointers
- Dynamic storage allocation
- Ability to call other Pascal programs
- FLEX version may call assembly language programs
- Buffered or single character terminal input
- Standard math functions: SIN, COS, ARCTAN, EXP, LN, SQR, SQRT
- Random number generator function
- Many usable, sample programs included
- UniFLEX version supports:
  - Random file positioning
  - Ability to call various UniFLEX system routines
  - Ability to execute UniFLEX utility commands

Pascal on diskette for 5" and 8" 6809 FLEX is available for $200.00 The 5" version requires two disk drives. The UniFLEX version is $300.00 and includes one year of maintenance. All orders should include 3 percent for postage and handling (10 percent on foreign orders).

™FLEX and UniFLEX are trademarks of Technical Systems Consultants, Inc.

technical systems consultants, inc.
Box 2570, West Lafayette, IN 47906
(317) 463-2502

'68' MICRO JOURNAL

Portions of text prepared using the following.

SWTPC 6800-6809-OMAF2-CDSI-CT82-Sprint 3
Southwest Technical Products
219 W. Rhapsody
San Antonio, Texas 78216

EDITOR - WORD PROCESSOR
Technical Systems Consultants, Inc.
Box 2573, W. Lafayette, IN 47906
FLEX is TM of TSC

GIMIX Super Maintrame-Assorted memory boards
GIMIX Inc.
1337 West 37th Place
Chicago, Il 60609

Publisher: Don Williams Sr.

Executive Editor: Larry Williams

Subscriptions and Office manager
Mary Robertson

General Girl 'Friday'
Joyce Williams

Contributing Editors:

Dr. Chuck Adams
Dr. Theo Elbert
Dr. Jeffery Brownstein
Dale Puckett
Russell Gore
Ron Anderson
John Jordon

December 1981
VOL III ... ISSUE XII

CONTENTS

Send All Correspondence To:

68 MICRO JOURNAL
5900 Cassandra Smith
Computer Publishing Center
PO Box 849
Hixson, TN 37343

615 842-4600

'68' Micro Journal is published 12 times a year by '68' Micro Journal, 6131 Airways Blvd., Chattanooga, TN 37421. Second Class postage paid at Chattanooga, TN. Postmaster: Send Form 3579 to '68' Micro Journal, PO Box 849, Hixson, TN 37343.

1-Year $18.50 2-Year $32.50 3-Year $48.50

—ITEMS SUBMITTED FOR PUBLICATION—

(Letters to the Editor for Publication) All 'letters to the Editor' should be substantiated by facts. Opinions should be indicated as such. All letters must be signed. We are interested in receiving letters that will benefit or alert our readers. Praise as well as gripes is always good subject matter. Your name may be withheld upon request. If you have had a good experience with a 6800 vendor please put it in a letter. If the experience was bad put that in a letter also. Remember, if you tell us who they are then it is only fair that your name 'not' be withheld. This means that all letters published, of a critical nature, cannot have a name withheld. We will attempt to publish 'verbatim' letters that are composed using 'good taste.' We reserve the right to define (for '68' Micro) what constitutes 'good taste.'

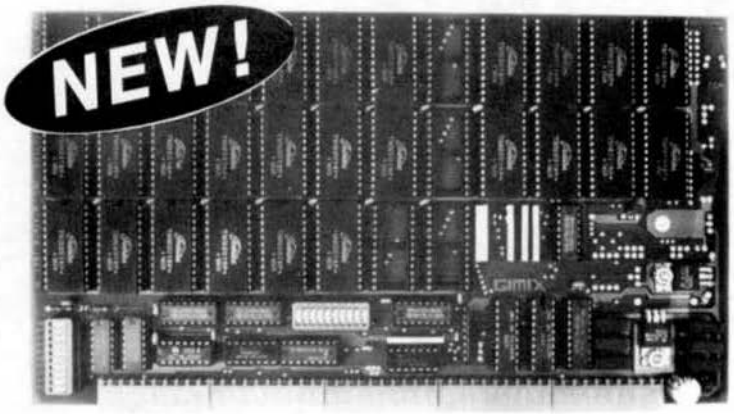
(Articles and items submitted for publication) Please, always include your full name, address, and telephone number. Date and number all sheets. TYPE them if you can, poorly handwritten copy is sometimes the difference between go, no-go. All items should be on 8X11 inch, white paper. Most all art work will be reproduced photographically, this includes all listings, diagrams and other non-text material. All typewritten copy should be done with a NEW RIBBON. All hand drawn art should be black on white paper. Please no hand written code items over 50 bytes. Neatly typed copy will be directly reproduced. Column width should be 3¼ inches.

(Advertising) Any Classified: Maximum 20 words. All single letters and/or numbers will be considered one (1) word. No Commercial or Business Type Classified advertising. Classified ads will be published in our standard format. Classified ads $7.50 one time run, paid in advance.

Commercial and/or Business advertisers please write or phone for current rate sheet and publication lag time.

64K BYTE STATIC RAM BOARD $638.67

for 6800 and 6809 systems using the SS-50/SS-50C bus



also available:

56K $578.57
48K $518.47
32K $398.37
24K $348.27

All versions have gold bus connectors and are fully socketed, assembled, burned in, and tested. Versions with less than 64K can be expanded at any time by adding additional RAM chips.

FEATURES:

★ ADDRESSABLE in two 32K sections with separate regular and extended address decoding for each section. Each section can be addressed to any 32K boundary in the address range (1M Byte with extended addressing). Each 32K section is divided into four 8K blocks that can be individually enabled or disabled. Disabled sections do not occupy address space.

★ FULLY STATIC MEMORY does not require complicated refresh timing or clocks for data retention. Compatible with any of the 6800/6809 DMA techniques.

★ GUARANTEED 2Mhz. OPERATION uses high speed (200 ns.) memory with no wait states or clock stretching required.

★ LOW POWER NMOS RAM requires less than 3/4 AMP (750 ma) typical at 8V, for a fully populated 64K board.

Also Available:

NON-VOLATILE 64K BYTE CMOS STATIC RAM BOARDS with BATTERY BACK-UP

With all the versatility of the above boards -- PLUS!

★ NON-VOLATILE MEMORY with built in battery back-up. Retains data even with system power removed. With the battery fully charged, data remains intact for a minimum of 21 days.

★ ULTRA-LOW POWER CMOS RAM requires less than 1/4 AMP (250 ma.) typical at 8V for a fully populated 64K board.

★ LOW BUS VOLTAGE DETECTION inhibits memory access during power up and power down to prevent false writes to the memory.

★ WRITE PROTECT SWITCH permits the entire board to be write protected for PROM/ROM emulation and software debugging.

64K . . . . . $988.64 **56K . . . . . $894.56** **32K $648.36**

NEW! **30 PIN SINGLE PORT SYNCHRONOUS AND ASYNCHRONOUS I/O BOARDS with selectable RS-232C, RS-423, and RS-422 Input/output.**

Versions available using:

★ 6850 ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER (ACIA) . . . . .
★ 6852 SYNCHRONOUS SERIAL DATA ADAPTER (SSDA) . . . . . } Contact **GIMIX** for information on pricing and availability.
★ 6854 ADVANCED DATA LINK CONTROLLER (ADLC) . . . . . .

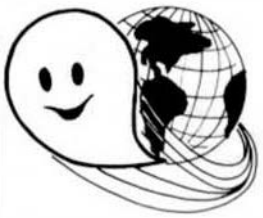
TO ORDER BY MAIL

SEND CHECK OR MONEY ORDER OR USE YOUR VISA OR MASTER CHARGE. Please allow 3 weeks for personal checks to clear. U.S. orders add $5 handling if order is under $200.00. Foreign orders add $10 handling if order is under $200.00. Foreign orders over $200.00 will be shipped via Emery Air Freight COLLECT, and we will charge no handling. All orders must be prepaid in U.S. funds. Please note that foreign checks have been taking about 8 weeks for collection so we would advise wiring money, or checks drawn on a bank account in the U.S.. Our bank is the Continental Illinois National Bank of Chicago, account #73-32033 Visa or Master Charge also accepted.

GIMIX INC. reserves the right to change pricing and product specifications at any time without further notice.

GIMIX* and GHOST* are registered trademarks of GIMIX Inc. © 1981 GIMIX Inc.

FLEX AND Unifax are trademarks of Technical Systems Consultants Inc. OS-9 is a trademark of Microware Inc. See their ads for other **GIMIX** compatible software.



GIMIX INC.

The Company that delivers

Quality Electronic products since 1975.

1337 WEST 37th PLACE, CHICAGO, IL 60609
(312) 927-5510 • TWX 910-221-4055

SEE GHOST AD PAGES 54, 59, 64

Does timesharing on a small system make sense? It does with OS-9 Level One!

Now two (or more) acts can share your microcomputer stage. You will no longer have to walk away from your computer while it is busy running a long program. Because OS-9 is a multitasking operating system, you can be running a BASIC program while editing a PASCAL program, for example. This lets you make more efficient use of your time and your system, even if you only use one terminal. If your application requires multiple, independent terminals, one OS-9 system can do the work of several single-user systems.

The convenience of an advanced operating system

Sophistication does not require complexity. Many OS-9 users say that it is actually easier to use than the older 6800-type operating systems. Consider how easy it is to run multiple programs; to run a program you just type its name and hit 'return.' To run a program as a separate job, you type its name, an '&' character, then hit return. The program runs as usual, but OS-9 comes back immediately and is ready for your next command. Simple commands let you see each program's status, set its priority, or abort it.

The file management system has fast, byte-addressable random-and sequential-access files. The tree-structured multiple directory system lets you create separate disk directories for each user, project, or application. Command line I/O file redirection means you specify what device and/or files a program will use when you run it, not when you write it.

Efficiency and hardware versatility

No other operating system can run on such a broad range of hardware: the overall RAM requirement for Level One is 32K to 56K RAM. Memory utilization is superlative because OS-9 lets multiple tasks "share" the same reentrant program. For example, if two users run BASIC09, only one "copy" is actually loaded into memory. The Level Two version of OS-9 can utilize up to a megabyte of memory on systems having memory management hardware (both versions come with complete timesharing support).

OS-9's device independent I/O system can handle almost any number and combination of I/O devices: five or eight inch diskettes, winchester disks, disk cartridges, serial and parallel ports, memory-mapped video displays, and more. Microware offers a large selection of "stock" device interface software modules, or you can create your own; all the information you need is in the manuals.

Excellent support and documentation

Each OS-9 package comes with a User's Manual and a System Programmer's Manual that cover every aspect of OS-9. If you have special requirements, you can even purchase the Source Code for most of OS-9 and related software. At Microware we take pride in offering the best customer support in the business. Technical advice and assistance by phone, mail or telex is available during all business hours.

Superb software tools

In addition to BASIC09, Microware offers: PASCAL, Interactive Assembler, Macro Text Editor, Stylograph, Word Processor, Interactive Debugger, and coming soon, COBOL, and C language compilers.

BASIC09 has a dual personality. One craves meat-and-potatoes BASIC. The other prefers Programme ala Pascal.

Some people say BASIC09 is really a PASCAL in disguise, others say it's still BASIC. You'll understand this delightful dilemma when you look at both versions of the "bubble sort" program shown below: both can be run by BASIC09. The program on top is unstructured and hard to understand, but it's traditional BASIC. The program on the bottom is well-structured and easy to follow, a virtue of PASCAL. With BASIC09 you can program either way, or mix the best of both. It's like getting two languages for the price of one.

SORT AN ARRAY
IN ASCENDING SEQUENCE

```
90   DIM A(5)
100  I=5
110  IF I=1 THEN 200
120  FOR J=1 TO I-1
130  IF A(J)<=A(J+1) THEN 170
140  T=A(J+1)
150  A(J+1)=A(J)
160  A(J)=T
170  NEXT J
180  I=I-1
190  GOTO 110
200  RETURN
```

```
DIM array(5)
outer=5
WHILE outer>1 DO
  outer=outer-1
  FOR inner=1 TO outer
    IF array(inner)>=array(inner+1) THEN
      temp=array(inner+1)
      array(inner+1)=array(inner)
      array(inner)=temp
    ENDIF
  NEXT inner
ENDWHILE
RETURN
```

Makes programs better

BASIC09 has five kinds of loop structures: WHILE . . DO, REPEAT . . UNTIL, LOOP . . ENDLOOP, FOR . . NEXT and IF . . THEN . . ELSE. If one of the five built-in data types (byte, integer, real, string, and boolean) doesn't suit the problem, you can make a new one of your liking with the TYPE statement. Need a tree, linked list, or symbol table? Complex non-rectangular data structures using any combination of data types are easy to define. Modular programming breaks down large programs to smaller, more manageable elements. BASIC09 or machine language recursion plus parameter passing to any other BASIC09 or machine language procedure. There is a complete set of statements for device-independent sequential or random I/O, plus a superlative PRINT USING system.

Makes programs faster

No full-feature BASIC for any 8-bit microprocessor is faster than BASIC09, because it is an interactive compiler. As each program line is entered, it is instantly compiled to a smaller, faster form. Because BASIC09 automatically converts programs back to original "source" form for listing, it is as friendly and easy-to-use as traditional interpreter BASICs. Each procedure can be independently compiled to position-independent, reentrant, ROMable format. Microware developed a new ultra-fast 9-digit-accuracy floating point math system just for BASIC09. And if that's still not fast enough, there's BYTE and INTEGER arithmetic.

Features that make programs easier to write

The compiler is integrated with a full-feature string AND line-number oriented text editor. If you make a mistake, BASIC09 tells you instantly. String-oriented commands such as search, change, change all occurances, delete, and insert can be used on programs with or without line numbers. There's an automatic line renumbering function too.

Features that make programs easy to test

Debugging often takes longer than writing a program. That's why BASIC09's integral high-level debugger sets it apart from all other compiled OR interpretive languages. The TRACE command shows you each statement executed in BASIC form, plus the result of any expression evaluation. STEP lets you run one or more statements at a time. LET and PRINT allow you to examine or change the values of variables, by name. STATE lists procedure calling order. And there are nine other debug commands. If you need to correct a program, you can edit, recompile, and rerun it in seconds.

Microware software is available for most popular 6809 computer systems.

Write or call for our free catalog. We accept phone orders and MasterCard and VISA orders.

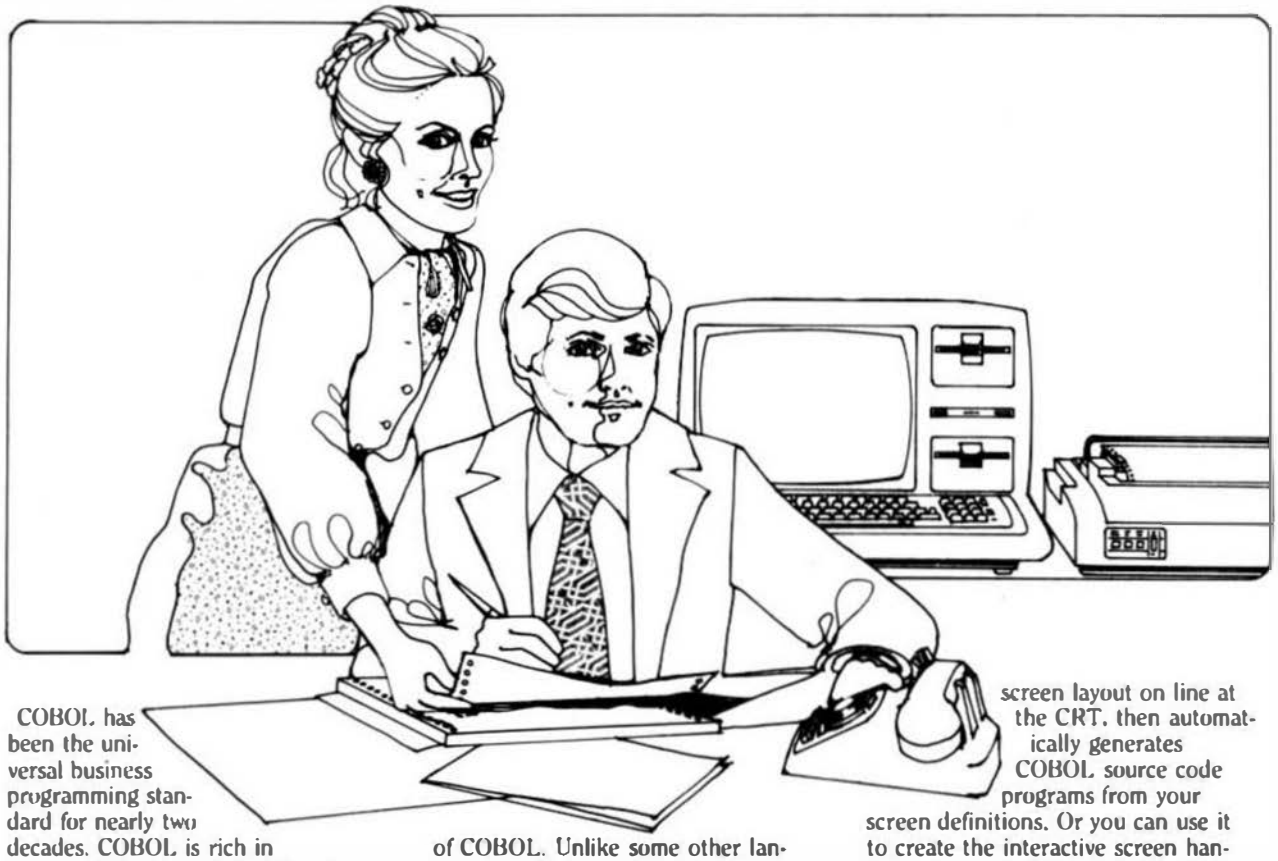
OS-9 is a trademark of Microware; BASIC09 is a trademark of Microware and Motorola.

MICROWARE

Microware Systems Corporation
5835 Grand Ave., Des Moines, IA 50312
(515) 279-8844 Telex 910-520-2535

CIS COBOL™:

The Natural Choice For Business Software



COBOL has been the universal business programming standard for nearly two decades. COBOL is rich in commercially-oriented facilities. It has powerful file handling, formatted printing, and data structure capabilities. It is English-like, so that programs are easy to read and maintain. By far, most organizations use COBOL as their main business programming language. One effect of this is that more professional business programmers know and use COBOL, and the best business application software is written in COBOL.

Microware has developed the 6809 version of Micro Focus's proven CIS COBOL compiler to allow you to run ANSI 1974 standard COBOL on your OS-9 based computer system. It's been certified as such by the U.S. General Services Administration, following stringent testing. This assures that CIS COBOL is compatable with standard minicomputer or mainframe COBOLs. And CIS COBOL has been proven on thousands of micro and mini systems all over the world.

Stability is an important advantage of COBOL. Unlike some other languages, a firm standard has been established. Because of this, COBOL programs can be transferred from one machine to another with a minimum of modifications. COBOL users can take advantage of the mass of existing programs written in COBOL.

CIS stands for Compact, Interactive, and Standard—the most desirable qualities for microcomputer COBOL. And CIS COBOL offers you much more! It has been specially designed for interactive operation and efficient use on small computers. CIS COBOL has multi-user capability that allows more than one COBOL program to be run simultaneously. CIS COBOL extensions for conversational applications, screen control, interactive debugging, and OS-9's device-independent I/O system.

CIS COBOL's optional FORMS 2 program generator eliminates the need to write simple data entry and inquiry programs. It lets you build a screen layout on line at the CRT, then automatically generates COBOL source code programs from your screen definitions. Or you can use it to create the interactive screen handling portions of more complex programs.

CIS COBOL and FORMS 2 can be used with any disk-based 6809 computer system having at least 48K of user RAM running Microware's OS-9 Level One or OS-9 Level Two operating systems.

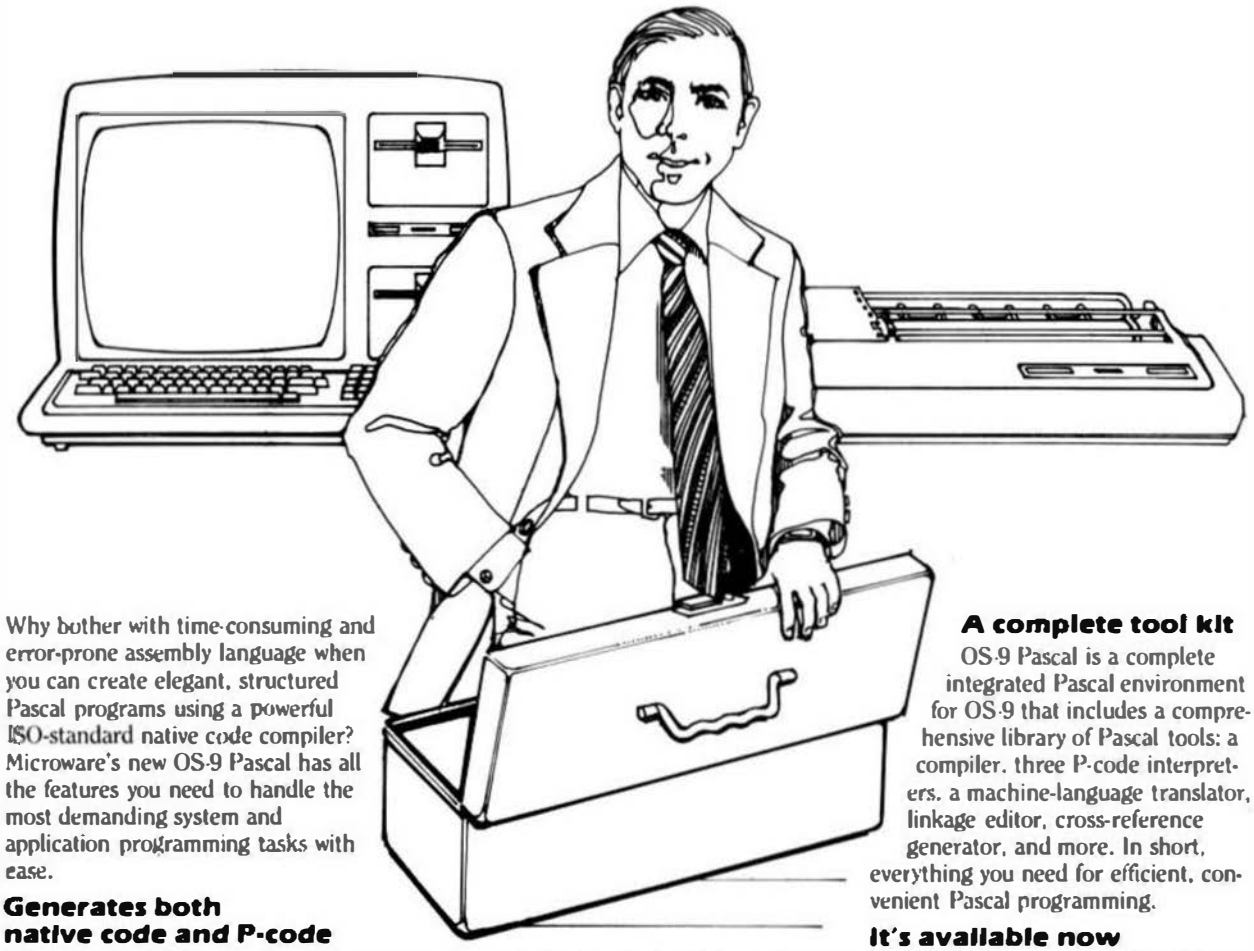
If you need to create business applications, COBOL is your natural choice. And if you want to run COBOL on your 6809 system--or want easy to use interactive business programming facilities—that means 6809 CIS COBOL.

Write or call for our free catalog. We accept phone orders and MasterCard and VISA orders.

MICROWARE®
Microware Systems Corporation
5835 Grand Ave., Des Moines, IA 50312
(515) 279-8844 Telex 910-520-2535

OS-9 PASCAL™:

A New Programming Tool For Experts



Why bother with time-consuming and error-prone assembly language when you can create elegant, structured Pascal programs using a powerful ISO-standard native code compiler? Microware's new OS-9 Pascal has all the features you need to handle the most demanding system and application programming tasks with ease.

Generates both native code and P-code

With OS-9 Pascal you don't have to make that difficult choice between easy-to-use P-code Pascal or fast native-code Pascal. You can compile your Pascal program to pure 6809 assembly language source code. OS-9 Pascal performs extensive local and global code optimization which results in incredibly fast and compact machine language programs. Or if you prefer, OS-9 Pascal can generate P-code for interpretive execution to simplify program debugging and testing. There's also a Virtual Memory P-code Interpreter that can run huge Pascal programs that other microcomputers can't touch. In fact, you can run programs using any combination of P-code, compiled machine language, or handwritten assembly language procedures.

ISO Standard Pascal Plus

OS-9 Pascal conforms to the ISO industry standard for Pascal, so you are assured of portability to or from any other computer that uses standard Pascal. OS-9 Pascal protects your software investment and gives you access to a vast body of existing Pascal software. Beyond the standard, we've added natural extensions to OS-9 Pascal to make it even more versatile, such as: relaxed identifier syntax; separate procedure compilation; random access file and interactive I/O; bitwise logical operators; run-time error handling; and much more. And because it runs under OS-9, it is inherently multiuser and multitasking.

A complete tool kit

OS-9 Pascal is a complete integrated Pascal environment for OS-9 that includes a comprehensive library of Pascal tools: a compiler, three P-code interpreters, a machine-language translator, linkage editor, cross-reference generator, and more. In short, everything you need for efficient, convenient Pascal programming.

It's available now

OS-9 Pascal is now available *off-the-shelf* in all OS-9 disk formats. It can be used on any disk-based 6809 computer running OS-9 Level One or Level Two. Each OS-9 Pascal package includes the compiler, machine language translator, P-code interpreters, run-time support packages, linkage editor, demonstration programs, and a comprehensive 120-page User's Manual.

Write or call for our free catalog. We accept phone orders and MasterCard and VISA orders.

MICROWARE

Microware Systems Corporation
5835 Grand Ave., Des Moines, IA 50312
(515) 279-8844 Telex 910-520-2535

TM **Dataman+**

DATABASE MANAGEMENT

ONE PROGRAM

THAT DOES IT ALL!

DATAMAN + combines the capabilities of DATAMAN, DATARAND and DATACALC. It has been redeveloped from the ground up after over 12 months of experience with the DATAMAN system. DATAMAN + is a powerful RANDOM Database Management system under FLEX, UNIFLEX and OS-9.

DATAMAN + is for BUSINESS systems with 56K user ram. You can use DATAMAN + for inventory control, work scheduling, mailing lists, sales reports and much more. The powerful report writer lets you create invoices, statements, form letters, and any other type of report your business needs. You can perform calculations with your data and print out the results. Special printer handling allows use of any size column output, not just 80 and 132.

DATAMAN + is password protected at the menu level so that redundent password prompts are eliminated. We've added the human touch with the use of the operators name and calculator style input. DATAMAN + checks for valid data types on input thus eliminating erroneous data in your database.

DATAMAN + 's report writer has added intelligence so that separate select programs need not be run to create different reports from the same database. As a matter of fact, the report writer is so flexible that you can use it to create invoices, statements, even form letters using data from the database. You can even perform calculations with the data and put the results in the report.

Setting up your system to run DATAMAN + is very easy and automatic. The entire system has been designed with the inexperienced user in mind. The operation of the system is so easy that although a manual is provided none is required to run DATAMAN + .

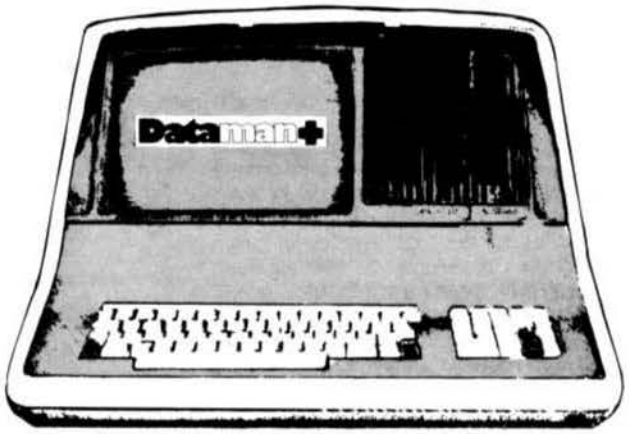
It's easy to create databases and reports with DATAMAN + . Full editing capability has been added to make it a snap. DATAMAN + is the first truly RANDOM DBM system to allow any size record and any number of fields.

Modifying DATAMAN + is easy because DATAMAN + comes with every line of source on disk! and an easy to use manual with sections on each program for the programmer who wants to make modifications or customize it.

An upgrade will be available for users with DATAMAN.

FLEX version available in December. $199.95
UniFLEX and OS-9 versions soon thereafter.

ONE PROGRAM
THAT DOES IT ALL!



FRANK HOGG LABORATORY, INC.

130 MIDTOWN PLAZA • SYRACUSE NEW YORK 13210 • (315)474-7856

COMING SOON
META X-FORTH
Create romable code from X-FORTH for any processor

OS-9 VERSION COMING

X-FORTH FLEX COMPATIBLE FORTH

By Charles (Chuck) Eaker, Ph.D

X-FORTH NEWS

COMING SOON . . .
OSBORNE GENERAL LEDGER in X-FORTH for FLEX and OS-9

This is the same G/L program that you usually see in BASIC but with the speed advantage of X-FORTH and of course runs much faster than the BASIC version. It does NOT require X-FORTH to run.

META-X-FORTH

This package will take an X-FORTH program and compile it into objectcode for any processor. This means that you can use X-FORTH to create programs for other computers. The code produced is romable.

OS-9 FORTH

We are taking X-FORTH and putting it on OS-9. This will mean that programs written in X-FORTH will run on both FLEX and OS-9 with minimal changes.

X-FORTH NOTES

If you are considering buying FORTH, then you are probably trying to decide which one of the two that are available for the 68XX to choose. Well, perhaps I can help by telling you some of the more major differences between the two.

X-FORTH runs in the FLEX (or OS-9) environment just like BASIC or any other FLEX program. The files that it uses are the same as any other FLEX program. This makes it compatible with other programs or utilities that you may have. The other FORTH is not. (see Ron Anderson's columns)

X-FORTH at $149.95 is more or less the same package that you get for $250.00 for the other FORTH. That is $100.05 less.

X-FORTH is faster, about 25% faster, although, exact timing tests haven't been run yet. The reason X-FORTH is faster is because we coded many of the important things in assembler, not high level FORTH.

X-FORTH documentation is without a doubt the best that's available for any FORTH on any computer. The manual is divided into four major sections. The first section (approx. 100 pages) is a very good tutorial on FORTH in general and gives the first time user a feel for the system. The second section (approx. 60 pages) goes into the extensions that were added for FLEX. The third section, the users manual (approx. 130 pages) is the part you will use the most. Suppose you wanted to work with strings, all you do is look in the section on string operations for all the information you would need. Each section explains in easy to understand terms how each word works and how to use it. The last section is the glossary. (approx. 66 pages) All the words described in the users manual are listed alphabetically with complete descriptions.

THE FUTURE

X-FORTH will be our major applications language in the future. Life is too short for BASIC. We are planning a complete business package in X-FORTH, A/R, G/L, A/P, etc. Because X-FORTH will run on many different operating systems, applications written in it will be much easier to maintain and of course the market is much bigger.

Well, I hope I've been able to answer many of your questions, but if you have more I'm just a phone call away.

Frank

Supplied on one 8" Disk or 2, 5" disk(s) with a 400 page manual in a hard cover binder. Disk(s) have the source of everything but the core of X-FORTH, which will be available later at extra cost. You get it all!!!! All for only

Manual available separately for $49.95, with credit for later purchase.

$149.95
+ $2.50

OS-9 Version Available

Exclusively From Frank Hogg . . .

DYNASOFT PASCAL FOR 6809 FLEX™

DYNASOFT PASCAL is now available for **FLEX 9™** systems!

Compiler Source Now Available

Price: Only $59.95. For run-time source add $30.00

For OS-9 add $10.00.

™Trademark of Technical Systems Consultants

BY JIM SCHREIER

NEW

THE BILL PAYER SYSTEM ™

THE BILL PAYER is a package of 10 menu driven programs in TSC Extended Basic. This powerful system helps you keep track of your bills. You can create a vendor list, enter invoices to be paid, generate reports about them, print your checks and much more. Uses random access files.

Explore Package now included at the same price.

THE PURCHASE ORDER system adds purchase orders to the BILL PAYER. This package of programs adds another level of control to your expenditures. Prints out purchase orders and keeps track of purchases. Requires the Bill Payer to work.

INCOME/EXPENSE LEDGER. This valuable package is most appreciated at tax time. Allows up to 99 income and expense numbers. Ties into the PURCHASE ORDER system, and the Bill Payer.

Includes manual and source supplied on disk in TSC Extended Basic.

THE BILL PAYER	$ 89.95
PURCHASE ORDER	49.95
INC/EXP. LEDGER	$49.95

BUY ALL THREE AT ONE TIME FOR 169.95 AND SAVE $19.90.

See Catalog on Page 9 for Ordering Information . . .

From Bud Pass

TABULA RASA

A VISICALC* TYPE PROGRAM

This program provides for the generation and maintenance of tabular computation schemes used for analysis of business, sales and economic scenarios. Simple user interface so even non programmer types can use it.

In TSC Extended Basic. Includes Source on Disk. $100.00

*T.M. PERSONAL SOFTWARE

See Catalog on Page 9 for Ordering Information . . .

FRANK HOGG LABORATORY, INC. 130 MIDTOWN PLAZA • SYRACUSE, N.Y. 13210
(315) 474-7856

NEW

By Dick Bartholomew

BASIC PROGRAMMERS TOOLKIT

$49.95 object only
$69.95 with source on DISK!

1. DECOMPIL — Turns BAC into BAS files
2. XREF — Cross reference listing of BASIC programs
3. EDIT — Edit a BASIC program that's in memory while in BASIC!!!!!

NOTE: For TSC XBASIC only

NEW

EXTENDED USE UTILITIES PACKAGE

$49.95 object only
$69.95 with source on DISK!

1. CRTSET like TTYSET for CRT's
2. USERINFO adds more than just diskname and number
3. SCAN list forward and backward thru a file
4. BROWSE like SCAN but in memory
5. LOAD alter loader
6. DISKDUMP to any part
7. INIT memory set command
8. SAVETEXT to disk
9. READTEXT from disk
10. REDIRECT change control part
11. REPLACE prompt memory change
12. MIRROR BACKUP fast sector copy

PASSWORD PROTECTION PACKAGE

$69.95 object only
$89.95 with source on DISK!

Six programs that allow you to protect your system by preventing a bootup without the proper password.

— Warning —

PASSWORD can be overcome if the user has access to another disk without password protection that will boot FLEX.

Programs are written in 6809 assembly language.

See Catalog on Page 9 for Ordering Information

From A.R. Ball

READTAPE

READ TRS-80 LEVEL II BASIC TAPES

This program, with an easy to make interface * will read TRS-80 LEVEL II BASIC tapes and convert the programs to TSC BASIC. Those things that can't be converted are flagged so that you can find them easily with the TSC text editor. Now you can use all that TRS-80 software out there.

6809 Assembly language - includes source on DISK! $54.95

*(instructions and schematic included - cost about $2 to build)

PLOT

Now you can have GRAPHICS added to all your programs. Just write the data out to a virtual array and call PLOT. PLOT is written in TSC XBASIC and the source is included on the disk.

INFINITE RESOLUTION GRAPHICS ON YOUR TERMINAL OR PRINTER. HISTOGRAMS, BARGRAPHS, XY PLOTS PLUS OTHERS.

IN TSC XBASIC. SOURCE INCLUDED ON DISK. $44.95

From Bud Pass

TABULA RASA

A VISICALC* TYPE PROGRAM

This program provides for the generation and maintenance of tabular computation schemes used for analysis of business, sales and economic scenarios. Simple user interface so even non programmer types can use it.

In TSC Extended Basic. Includes Source on Disk. $100.00

*T.M. PERSONAL SOFTWARE

SUPER SLEUTH

Disassembler System

Examine and/or modify object program files on disk or in memory for 6800, 6801, 6809 ..... $99.00

SUPER SLEUTH Z-80-8080-8085 Version of above $99.00

CROSS-ASSEMBLER for 6800/1, 6805, 6502, Z-80, 8080/5 ..... ea. $50.00

Spelltest

From DALE PUCKETT

Correct those misspelled words with SPELLTEST. FAST 6809 MACHINE LANGUAGE

SPELLTEST reads in text, tests it for misspelled words, and then lets you correct them. It then writes the corrected file back to disk. You do NOT have to EDIT the file!!! All the other spelling programs flag the words and you have to edit the file. SPELLTEST does the work for you!

SPELLTEST's other features include full prompting, output to terminal or printer, capability to use several dictionaries in a variety of ways, as well as allowing you to add to the dictionary. Plus much more.

FAST...reads and tests a 3500 word text (about 10 pages) in less than a minute!! SPELLTEST is first with an INTELLIGENT DICTIONARY!

$199.00 source available for $100 additional

Comparable to CP/M programs costing $295.00.

Buzztest

Buzzword tester coming soon

Readtest

Readability tester

$74.95 with source $54.95 without

SOFTWARE CATALOG

PROGRAM	LANGUAGE	OBJECT	w/ SOURCE ON DISK
X FORTH	6809/6800		---$149.95
*Bill Payer	TSC XBASIC		89.95
*Purchase Order	TSC XBASIC		49.95
*Income/Expense	TSC XBASIC		49.95
(*)All Three	TSC XBASIC		169.95
Basic Prog. Toolkit	6809 ASMB	$49.95	69.95
Password Protection	6809 ASMB	69.95	89.95
Extended Utilities	6809 ASMB	49.95	69.95
Job Control Prog.	6800/6809 ASMB	69.95	89.95
Esther	6800/6809 ASMB	39.95	59.95
Readtest	6800/6809 ASMB	54.95	74.95
Help	6800/6809 ASMB	29.95	49.95
Dynasoft Pascal	6809	59.95	** 89.95
Plot	TSC XBASIC		44.95
Read TRS80 Tapes	6809 ASMB		54.95
Super Sleuth	6809/6809		99.00
Z80 Super Sleuth	6809/6809		99.00
Cross Assemblers	MA ROS FOR TSC 6809 ASMB		EA. 49.95
	6800/1, 6805, 6502, Z-80, 8080/5	3 for	99.95
Mailing List	TSC XBASIC/6809		99.95
Forms Display	TSC XBASIC/6809		49.95
Tabula Rasa	TSC XBASIC/6809		100.00
Fuzz/Corr/Study	TSC XBASIC/6809		400.00
PAY DAY	TSC XBASIC		195.00

** SOURCE AND REPRODUCTION LICENSE RUNTIME ONLY

***Includes everything but the core

U.S.A. add $2.50 for Standard UPS Shipping & Handling
Foreign orders add 10% Surface, 20% Airmail.

Specify 5" or 8" size disk and if for 6800 or 6809 system.

OUR SOFTWARE IS GIMIX *COMPATIBLE.*

OS-9 VERSIONS TO COME.

FRANK HOGG LABORATORY, INC.

130 MIDTOWN PLAZA • SYRACUSE, N.Y. 13210
(315) 474-7856

Flex User Notes

Ronald W. Anderson
3540 Sturbridge Court
Ann Arbor, MI 48105

GIMIX 58 DISK CONTROLLER

I've recently been having lots of trouble reading 5 inch disks from software suppliers on my old Shugart SA-400's. I decided that perhaps the trouble was in my old DC-1 controller card, and scrounged up a used DC-2. I found that I could read disks with it that were not readable with the DC-1, but I still got "re-trys" in several places on some of the disks. Plugging in the GIMIX board and connecting my 5" drives to it allowed me to read all 4 disks that I had previously had difficulties with, without a single re-try! GIMIX indicates that they have done two things that make this board a better disk reader. They have designed the circuit to meet the data hold-time requirements of the Western Digital 1771 controller IC, and they have not used the 1771's built in "data separator", but rather have an external one that is more complex and that works better.

The 58 board has some other nice features. When used with GIMIX FLEX on a 6809 system, it will control up to 4 disk drives in any combination of 5" and 8", with single or double sided disk access. It uses a 1771 controller, and therefore will not support double density. (it will support double TRACK density and double sided disks when used with GIMIX FLEX). I'm impressed with what it does. It fits the I/O bus, since it is a "programmed I/O" device and not a DMA controller. It is available for $226.58 as described here, and for $198.48 for the 5" drive only version. If you are going to run only 5" drives such as the MF-68, without the fancy's of double sided or double track density, this card is a direct, plug-in replacement for the DC-X series controllers. Under these conditions, it will run with standard FLEX9 as supplied by SWTPC and TSC (and FLEX2 in a 6800 system). It will also run in a 6809 system with 8 inch drives and a DMA controller, by using the USEMF utility. In this configuration, the 8" drives become 0 and 1, and the 5" 2 and 3.

A double density version, the 28 is also available, and it allows double density in addition to all the features described above. The price is $348.28. Most impressive feature of the board I tested was the ability to read disks that my original DC-1 found unreadable, and to do it without so much as a single re-try.

MORE ON WHEEL RE-INVENTION

I hope you won't think I am harping on something, but creativity is sort of a pet subject of mine. A recent article in our local Ann Arbor News was a story about George Muller, a local man who, in the past 30 years has accumulated 87 patents. Mr. Muller is presently Technical Planning and Features manager for car engineering for Ford Motor Company. Mr. Muller has been holding workshops and seminars since 1968 for the American Management Association. Part of what Mr. Muller tells people in his seminars is that they shouldn't go to the Libraray first and find out how other people have solved the problem, but rather to see what they can come up with on their own.

Of course 98 per-cent of the time, such an approach leads to re-discovering some solution to a problem, but occasionally a new approach to a solution is discovered. It just happens that I have my name on a couple of patents too. Both were the result of my sitting down with a pad and pencil and thinking about the problem. At the time I lived 250 miles from Ann Arbor, the location of the company for which I was consulting. Neither patent was the result of finding out how other people had solved the problem at hand. I suspect that knowledge of other solutions would have prevented the generation of new approaches to the problems. On the other hand, I once applied for a patent for what i thought was a novel decade counter circuit (back in the days of discrete transistors). This circuit used only 9 transistors, and drove a gaseous discharge display tube. The previously used circuit had 19 transistors and considerably more other components to do essentially the same thing. After a year or so, the patent examiner indicated that Bell Labs had a patent on one of the parts of my circuit, and ITT had the other part patented. Both patents were very recent. Of course I got philosophical and decided that I was in pretty good company with the idea anyway.

MINIFLEX USERS TAKE NOTE

I received a letter a few weeks ago from J. Preston Brashear, III. Preston is a MiniFlex user who wants to do something to get that group organized. He will, by the time you read this, have contacted any of you who have written letters or articles indicating interest in MiniFlex for '68' or other magazines. My contribution has been all the material in my first 8 newsletters, published independently, before I started this column for '68' Micro Journal. At that time I had MiniFlex running, and many of the articles and programs are MiniFlex versions. Preston has indicated a willingness to distribute re-prints of these 8 newsletters. If you are interested in these and/or other aspects of MiniFlex, you may contact Preston by writing him. His address is 1580 Eastgate Drive, Suite 320, Garland, TX 75041. Some of the projects he has in mind are, the distribution of information regarding any MiniFlex compatible software available, and documentation of patches to FLEX2 compatible programs to make them MiniFlex compatible. I'll do anything i can to help in this project.

CORRESPONDENCE PROBLEMS

A while ago, I indicated that I had tried at least, to answer all correspondence that comes to me. I'm sure that some has fallen in a crack somewhere and not gotten answered at least directly. Recently, things have been getting out of hand in terms of the number of letters that arrive here every week. I find that I no longer have time for individual replies to all the letters I receive. As undemocratic as it may seem, I will answer letters as I can, but won't guarantee a try at all of them. Lately I have been going through a roll of 100 stamps (now $18.00 a roll) at an ever increasing rate. If you want a personal reply, a stamped self addressed envelope will insure some kind of a reply, be it several words scribbled on a piece of computer paper or 6 pages typewritten. If I get too wordy for one stamp, I'll supply the extra one!

This column is a "spare time" activity. I presently spend more hours answering correspondence than I do writing the monthly column. That is not a complaint, for if all the correspondence stopped, I would have a rather hard time finding things to write about. Please, however, be patient in waiting for a reply.

MORE ON BUS TERMINATION

After my short note on intermittent problems in which I mentioned bus terminators, I received a letter from Terry Ritter, PE containing a two page dissertation on computer bus lines, terminations and TTL devices. I will quote portions of it here. "TTL devices do not work best when "terminated with a proper load"; they work best when terminated with NO load at all. Any load, be it pull-up, pull-down, or pull-to-center requires the driver to source (or sink) additional current over the no-load condition. This

results in reduced logic levels (although the actual amount of the reduction may or may not be significant), and reduces the noise margin of the system."

Mr. Ritter goes on to point out that the terminations add a ground current component of about 300 ma that varies with the states of the buses at any given time. He points out that this variation in ground current could reduce noise margins on some boards. He goes on to indicate that a well designed bus has considerable capacitance to ground, and that a logic level on one of these busses therfore represents considerable stored energy. When the bus driver goes into a high impedance state (tristate driver), the data on the bus may remain valid for several bus cycles, because of the capacitance. "While it would be inadvisable to rely on this effect for long periods, it is useful to assure that data remains valid on the data bus after the falling edge of (the) E (clock). This may be important, since clock delay to other boards may result in their latching the data from the data bus after the CPU board has ceased to drive the bus. Inherent data bus hold time protects against such clock "skew"; data bus "terminations" destroy that protection.

Mr. Ritter continues, pointing out that if the busses are considered as transmission lines, one finds that reflection from a high impedance (open) end will reflect IN PHASE with the signal, resulting in BETTER logic levels. When the reflected wave again encounters the driver, a low impedance there will cause an out of phase reflection that will reduce the logic level. This implies that the best termination is not a bus end termination, but a series termination at the driver. He sums up his letter by saying "In short, your proposed "termination" provides no benefit, is basically the wrong way to terminate logic busses, destroys data bus hold time, wastes power, reduces noise margins, and perpetuates an electronic myth."

All I can say, is that the arguments presented sound very logical. Would someone care to submit the arguments in favor of terminating the bus? I might add to Mr. Ritter's comments a reminder that TTL logic levels are defined such that a voltage level below 0.8 is a logic 0, and levels above 2 volts are a logic 1. 1 levels higher than 2 volts, and 0 levels lower than 0.8 volts are better in that some noise voltages on the bus may drive it toward the logic threshold (which is somewhere between 0.8 and 2 volts). If it is farther from that threshold, more noise voltage will be tolerated before a false logic level is produced. If termination simply results in a less ragged looking bus waveform at the expense of reducing the logic levels, it probably has made the noise margins lower. My best guess would be that those terminations that seem to be most used will reduce the 1 logic level considerably.

I've spoken to a few of the advocates of bus termination, with the following information having been gleaned from the conversations and my previous knowledge of the subject. Whether a data bus should be considered a transmission line in the classical theoretical sense, depends on the physical length of the bus and the frequency of the signals that are to be transmitted on it. Transmission lines are best characterized by their length in "wavelengths" at the frequency to be transmitted on them. A system has at least one, and frequently more clock signals on the bus. These signals are usually in the form of a "square wave". A square wave contains frequency components that are many times the fundamental clock frequency; a fact that may be clearly understood if you have noted any interference in your television reception on the lower channels. These channels are below the standard FM broadcast frequencies which start at 88 mhz. Channels 2 through 6 occupy the area from approximately 60 to 88 mhz. It should be apparent that there are frequencies present at or above 60 mhz in your system. At 100 mhz, the wavelength of a radiated signal in air is 3 meters. Since the propagation velocity of a signal along a transmission line is somewhat less, the wavelength might well approach 2 meters.

A signal conductor must be treated as a transmission line if its length approaches a significant fraction of a quarter wavelength. At this length, a transmission line acts like a transformer. An open or unterminated end is transformed at the other end of the line so that it looks like a short circuit, a short circuit at the far end looks like an open circuit at the near end, and no signal may be transmitted at the frequency where the line is exactly one quarter wavelength. If the line is not exactly 1/4 wave, but nearly so, a high impedance at one end is transformed to a low impedance at the other. The problem is not very significant for our 1 mhz systems that have a data bus length of 6 inches or so, but in systems that operate at 2 mhz to 8 or 10 mhz (as in 68000 systems), and have 15 card slots so that the length of the bus approaches half a meter, the problem can become considerable.

I guess the real point of all this discussion is that the choice of termination or no termination is not trivial. Any given system or backplane may be designed to work either terminated or unterminated. Haphazzard application of terminators to a system that was designed to be unterminated, and vice-versa may be a disaster. Beware.

INTERMITTENT PROBLEMS AGAIN

I received a letter from Al Moreira, who indicated that my reported heat problem induced him to remove the cover from his mainframe and that the results were the same as mine, an immediate drop in the disk read error rate. Al mentioned that he is installing a fan in the mainframe as I have done. I don't think I had mentioned earlier that I had had considerable problems very early, with my MF-68 disk drive. I solved most of them by using a "nibbling tool" to remove most of the rear of the MF-68 chassis, leaving only enough for adequate heat sinking of the regulators. That means that the area behind the drives is wide open. All my early troubles disappeared after this modification. Al also mentioned that he is in agreement with my remarks regarding the desirability of spending more time writing smarter programs and less worrying about processor relative speeds. I might add that I am in agreement that comparisons of processors and compilers only make sense when one uses the same algorithm for all the tests.

HOBBYIST COMPONENTS GO INDUSTRIAL

I recently finished the design of the electronics for a balancing machine. Since a competitor will undoubtedly read this, I can't say a great deal about the software or the application. However, I can indicate that I have saved many thousands of dollars by building the prototype, and probably the first 25 or so, production units around available components. The backplane is from Thomas Instruments. The processor board is a MP-09 from SWTPC, with 2K of ROM and 2K of 2716 compatible RAM plugged into it. A MP-S board drives a 40 character electroluminescent display from Digital Electronics. The display accepts 1200 baud serial data. A JPC 16 channel Analog to Digital converter card handles input signals and also senses the position of switches and potentiometers that act as front panel controls for the balancer. A MP-LA serves as input for further switches, and drives a solid state relay for a "part balanced" indicator light. I've also used the Thomas transition board that does the address decoding for the I/O board and ties it to the main 50 pin bus. The program just fits a 2K EPROM, and 2K of RAM is more than adequate for the program and the 6809 stacks.

MORE ON PRIMES

I received a letter recently from Doug Beck, who uses a Motorola Development system that runs a 68000. Doug has Motorola Pascal for the 68000, and reports that program I presented in July '68' finds the primes to 10000 in 31 seconds. Since TSC Pascal does the same in 57, I am a bit disappointed. That means that a 2 mHz 6809 would beat the 68000, which, though Doug didn't elaborate, would probably be running at 8 mhz. I sincerely hope the Motorola Pascal is the culprit. Early on, I found Motorola BASIC for the Exorciser system (6800) to be the Ulterwyk version that we all remember as very good but no speed demon. Motorola's Fortran was only about 3 times faster than Ulterwyk BASIC. Perhaps the Motorola Pascal is not a true indication of the speed to be gained by using the 68000 (I hope).

6809 CONVERSION PROBLEMS?

My "printer pal" Art Weller reports a case of "stupidity" in converting his print routine from 6800 to 6809. Art reports that his print routine for Flex2 was "packed into the allocated space as tight as it can get." When he re-Orged it and assembled it with the 6809 assembler, it bombed. Seems that (as we all know, including Art), the 6809 assembler "expands" the byte count a bit because it has to simulate some of the 6800 instructions. Art's PINIT section spilled over into the PCHK section and wiped it out. Of course, we all know that a little "6809izing" of the source code will shrink the result generally to less bytes than the 6800 version. However, in our eagerness to get something new going, we can have lapses of memory and induce mysterious difficulties for ourselves.

SCATTER Plot

Buren R. Shields
900 Idlewilde LN. SE
Albuquerque NM 87108

The BASIC program in this article will produce scatter plots of data contained in data statements at the beginning of the program. It is written in SWTPC 8K basic but users should have little difficulty adapting it to their system. Only three functions are used. INT(X), TAB(X) and CHRS(X). A fourth function, FNA(x), is defined in the program and rounds numbers to the nearest integer value. If your BASIC has a rounding function, FNA(X) can be replaced with your function and statement 0120 dropped.

The program has been written for a system where the printer is on output port No. 3. If your printer is on another port modify the print output statements.

The data to be plotted is entered as DATA statements. The first quality in the data must be the number of observations. It is followed by the data pairs in the form A(1), B(1), A(2), B(2).....,A(N), B(N). Your DATA statements should replace the DATA statements, 0060 to 0090, which contain the data used in the sample program run.

A sample of the program output is shown. The scatter plot shows how the closing weekly prices of Abbott Labortories common stock varied as a function of time between the dates 5 February to 4 June 1979. The actual dates were replaced by the numbers 1 through 18 as variable B. Cost per share is plotted as variable A along the Y (vertical) axis and the number representing the week, as variable B, along the X (Horizontal) axis.

The program is designed so that the values along each axis range from the minimum to the maximum values found for the variable found along that axis.

This distributes the plotted values throughout the total area bounded by the two axes. Sometimes, this results in some unusual values being asssigned to the intervals of the plot. To assist in determining the coordinates of the plotted points, the values for each line of the Y (vertical) axis are printed out by the program. Statement 0400 sets the number of digits printed to the right of the decimal point. If the values to be plotted are very large or very small this statement will have to be changed accordingly.

Only three values are printed along the X axis. There was not room to print more than a few values. The increments for each axis unit are printed below the plot so interpolation of coordinates should not be too difficult. The program prints a number at each point plotted on the scatter plot. This number indicates the number of observations that have these same coordinates. The maximum number that can be indicated is nine. In the sample printout only one observation occured at each plotted point so only the number "1" was printed.

The program prints a listing of the data after the scatter plot. This list has been sorted into the order necessary for plotting. If the quantity of data

is very large and the user wishes to omit the listing, replace statement 0920 with a "STOP".

I hope that you enjoy this scatter plot program.

```
0010 REM SCATTER PLOT PROGRAM
0020 REM WRITTEN BY WILLIAM P. SHIFLET
0030 REM NUMBER OF OBSERVATIONS AND DATA TO BE PLOTTED
0040 REM ENTERED AS DATA STATEMENTS
0050 REM IN THE FORMAT N,A(1),B(1),A(2),B(2),...A(N),B(N)
0060 DATA 10,32.450,1,31.600,2,31.000,3,31.000,4,30.000,5
0070 DATA 31.125,6,31.500,7,32.695,8,32.875,9,33.250,10,
0080 DATA 32.875,11,32.375,12,32.125,13,31.375,14
0090 DATA 32.000,15,33.000,16,32.625,17,31.750,18
0100 PRINT #3
0110 REM DEFINE ROUNDING FUNCTION
0120 FOR I=1 TO 41 STEP 2
0130 DEF FNA(X)=INT(X)+INT((X-INT(X))*2)
0140 REM READ IN NUMBER OF OBSERVATIONS
0150 PRINT #3,TAB(22);"SCATTER PLOT"
0160 PRINT #3
0170 PRINT #3
0180 READ N
0190 DIM A(N,2)
0200 REM READ IN DATA
0210 FOR I=1 TO N
0220 FOR K=1 TO 2
0230 READ A(I,K)
0240 NEXT K
0250 NEXT I
0260 REM SORT DATA ON VARIABLE A
0270 GOSUB 1140
0280 REM OBTAIN MIN & MAX VARIABLE B VALUES
0290 GOSUB 1030
0300 REM CALCULATE AXIS SCALING FACTORS
0310  Y1=20/(A(1,1)-A(N,1))
0320  X1=40/(B1-B2)
0330 REM INHIBIT LINE LENGTH CONTROL
0340 LINE- 90
0350 REM LABEL Y-AXIS
0360 PRINT #3,"VAR A"
0370 PRINT #3
0380  K=1
0390 FOR I=20 TO 0 STEP -1
0400 REM SET # PLACES TO RIGHT OF DECIMAL =
0410 DIGITS= 3
0420 REM PRINT Y-AXIS VALUES
0430 PRINT #3,TAB(1);A(N,1)+1/Y1*TAB(K));I-1
0440 REM REMOVE DECIMAL LIMITATION
0450 DIGITS= 0
0460 REM CHECK FOR END OF DATA
0470 IF K=N+1 THEN 730
0480 REM SET POINT COUNTER
0490  L=49
0500 REM IS NEXT POINT ON THIS LINE?
0510 IF I=FNA((A(K,1)-A(N,1))*Y1) THEN 540
0520 GOTO 730
0530 REM LAST POINT IN DATA SET?
0540 IF K<N THEN 580
0550 PRINT #3,TAB(8+FNA((A(K,2)-B2)*X1));CHR$(L)
0560 GOTO 740
0570 REM NEXT POINT ALSO ON SAME LINE?
0580 IF I=FNA((A(K+1,1)-A(N,1))*Y1) THEN 630
0590 PRINT #3,TAB(8+FNA((A(K,2)-B2)*X1));CHR$(L)
0600  K=K+1
0610 GOTO 740
0620 REM NEXT POINT HAVE SAME X COORDINATES?
0630 IF FNA((A(K,2)-B2)*X1)=FNA((A(K+1),2)-B2)*X1) THEN 690
0640 REM NO, PRINT DATA POINT, STAY ON LINE
0650 PRINT #3,TAB(8+FNA((A(K,2)-B2)*X1));CHR$(L);
0660  K=K+1
0670  L=49
0680 GOTO 540
0690  K=K+1
0700  L=L+1
0710 IF K=N THEN 550
0720 GOTO 630
0730 PRINT #3
0740 NEXT I
0750 REM PRINT LOWER Y-AXIS VALUE
0760 FOR I=1 TO 41 STEP 2
0770 REM PRINT X-AXIS
0780 PRINT #3,TAB(7+I);"+"
0790 NEXT I
0800 PRINT #3
0810 REM PRINT X-AXIS VALUES
0820 PRINT #3,TAB(3);"I";TAB(9);I-1;TAB(41);"I"
0830 PRINT #3,TAB(8);B2;TAB(28);(B2+(B1-B2)/2);TAB(40);B1
0840 PRINT #3,TAB(5),"VAR B"
0850 PRINT #3
0860 PRINT #3
0870 PRINT #3
0880 PRINT #3,"AXIS SCALING FACTORS"
0890 PRINT #3
0900 PRINT #3,"INCREMENT PER Y (VERTICAL) AXIS UNIT = ";1/Y1
0910 PRINT #3,"INCREMENT PER X (HORIZONTAL) AXIS UNIT = ";1/X1
0920 PRINT #3
0930 PRINT #3
0940 PRINT #3,"VARIABLE LISTING"
0950 PRINT #3,"VAR A","  VAR B"
0960 FOR I=1 TO N
0970 FOR K=1 TO 2
0980 PRINT #3,TAB(K),A(I,K);
0990 NEXT K
1000 PRINT #3
1010 NEXT I
1020 STOP
1030 REM SUBROUTINE TO OBTAIN MIN
1040 REM B1=B MAX, B2=B MIN
1050  B1=0
1060  B2=99999999
1070 FOR I=1 TO N
1080 IF A(I,2)<B1 THEN 1100
1090  B1=A(I,2)
1100 IF B2<A(I,2) THEN 1120
1110  B2=A(I,2)
1120 NEXT I
1130 RETURN
1140 REM SORT SUBROUTINE
1150  F=0
1160  N1=N-1
1170 FOR I=1 TO N1
1180 IF A(I,1)>A(I+1,1) THEN 1200
1190 GOTO 1290
1200  T1=A(I,1)
1210  T2=A(I,2)

1220  A(I,1)=A(I+1,1)
1230  A(I,2)=A(I+1,2)
1240  A(I+1,1)=T1
1250  A(I+1,2)=T2
1260  F=1
1270 GOTO 1280
1280 REM DOES NEXT A VARIABLE HAVE SAME VALUE?
1290 IF A(I,1)=A(I+1,1) THEN 1310
1300 GOTO 1380
1310 REM SORT ON VARIABLE B VALUES
1320 IF A(I,2)>A(I+1,2) THEN 1340
1330 GOTO 1380
1340  T2=A(I,2)
1350  A(I,2)=A(I+1,2)
1360  A(I+1,2)=T2
1370  F=1
1380 NEXT I
1390 IF F=1 THEN 1140
1400 RETURN
```

```
                    SCATTER PLOT

VAR A

32.750 :                    I
32.562 :
32.375 :
32.187 :
32.000 :                                I
31.812 :                 I    II
31.625 :               I                 I
31.437 :                      I
31.250 :                           I    I
31.062 :
30.875 :                                I
30.687 :                      I
30.500 : I          I                   I
30.312 :                           I    I
30.125 :          I
29.937 :     I I
29.750 :
29.562 :
29.375 :
29.187 :
29.000 :         I
       :  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
       :  I                    I                    I
       :  I                    0.5                  1
                  VAR B
```

AXIS SCALING FACTORS

INCREMENT PER Y (VERTICAL) AXIS UNIT = 0.1875
INCREMENT PER X (HORIZONTAL) AXIS UNIT = 0.02500000

VARIABLE LISTING

VAR A	VAR B
33.75	10
33	16
32.875	9
32.875	11
32.75	1
32.625	8
32.625	17
32.375	12
32.125	13
32	15
31.75	18
31.5	2
31.5	7
31.375	14
31.125	6
31	3
31	4
30	5

* * COLOR COMPUTER *Users Notes* * *

by Robert L. May
Rt 7 Box 290A, Gadsden, Al. 35903

INTRO:

This month we'll continue to present "reviews", with a little different twist thrown in) a look at a SUPER Book. We get our first look at "the other Disk System" (Non Radio Shack) for the Color Computer, and look at CER-COMP's excellent software. First, we'll look at a Basic BASIC Program Monitor.

REVIEW:

Armadillo Software
P.O. Box 7661, Austin, Tx. 78712

ARMADILLO SUB — A Machine Language Monitor, written in BASIC
For 16K Color Computers — does NOT require EXTENDED BASIC

"ARMADILLO BUG" is a BASIC Program which provides a functioning Monitor for the COLOR COMPUTER. It is a 200 statement program providing the primary "Machine Language" Monitor Commands of <M>emory examine/change,<G>o-to a memory location to execute a program located there, <MOVE> a block of data from one location to another, <F>ill an area of memory with a specified value, <P>unch a mach. lang. program to tape, <L>oad a mach. lang. program from tape, and a <Z> command which clears the screen. All commands are single-letter except the "MOVE" command, and all address entries are expected as hexadecimal numbers. "ARMBUG", the program filename, provides a menu when it is RUN, and features very complete "entry prompting". For example, to use the "FILL" routine from the Menu:

```
type in F <ENTER>
BEG ADDR OF FILL    0400 <ENTER>
END ADDR OF FILL    05FF <ENTER>
VALUE OF FILL       FF   <ENTER>
```

and watch the TV Screen become Orange. Since it runs in BASIC, you can watch what is happening, rather than have the Screen become instantly Orange. The Program also allows continuous working in the "M" (Memory examine/change) Command Mode with the use of the + or - to advance to the next byte, or - to back up a byte.

The Documentation included with the tape is extremely WELL DONE. Since this is obviously a Program for the new Computer User, the Users Manual is very thorough and leaves no doubt as to what the program will do or how to use it. The examples provided are normally Screen Oriented, so the Operator can see what is happening, leading to a quicker insight of the operation of the Commands.

Another thing to consider, for the "less experienced" computer users, is that, since this IS a BASIC program, it can be listed and studied for the purpose of gaining more insight into programming with BASIC. You can also play around with changing commands to see what happens, etc. --- Just be sure you keep a good ORIGINAL copy handy, if it "blows up in your face, or bombs out," shut the Computer down for about 20 secs and reload the program and try again. The program is not "compressed", i.e., putting several statements in one line, so will be a valuable learning aid as well as a good functional program.

All in all, an excellent "basic" monitor "BASIC" program.

QUICK LOOK:

TALLGRASS TG-99 DISK INTERFACE and
TG-34 FLEA MONITOR SYSTEM

(will work with 4K, does not require or use EITHER BASIC ROM)

Pricing -- TG-99 Disk Interface, $99.95; TG-34 FLEA Mon. Sys. $34.95.

TALLGRASS Technologies Corp.
P.O. Box 12847
Overland Park, Ks. 66212

The TALLGRASS Color Computer Disk Interface is the second entry in non - Radio Shack Disk Controller market. Where the Chatron system we looked at last month is aimed at providing OS-58 Bucs (that is, FLEX, Trademark of TSC, and OS-9, Trademark of MICROWARE) and Radio Shack Operating System compatability while remaining price competitive with the R.S. Disk System, TALLGRASS has opted for very high storage capacity per Disk at a relatively low price. This is accomplished thru the use of TTL Circuitry instead of a Disk Controller Chip, and the use of the GCR (Group Coded Recording) encoding and decoding techniques. GCR encoding is a process which converts half of a byte (a nibble), or four bits, into five bits for storage on the Disk. Now, you ask, can converting 4 bits to 5 bits end up as "double density"? It IS a little confusing! Standard "Single Density", FM or Frequency Modulation, stores data, clock pulse, data, clock pulse, data, clock pulse, etc. In other words, every other pulse, or "storage location", is NOT a data pulse. Half of the storage locations on a disk are taken up with non-data storage, in this case, timing information. Standard "Double Density" storage, MFM or Modified Freq. Mod., basically stores a 'one' as a data bit in a storage location, and if the data bit is a 'zero', stores a clock pulse. Since every storage location represents a data bit, instead of every other storage location, you get "double density". It's rough, but thats the general idea. Now, back to "GCR". The converted 'nibble', which is now 5 bits, is stored similar to the MFM technique, where each storage location defines a bit of data. Effectively, then, you are eighty close to the "standard" double density. Also, the 4 bits can represent 16 unique digits, 0 thru F; while the 5 bits can represent 32 unique values, 0 thru 37. This means that only half of the possible combinations of 5 bits that were stored on the diskette can be "valid" data. The chances are good that a bad "write" to the disk, or a dirty spot, or any other error-producing condition, will yield an 'invalid' bit combination, which will be detected when decoded. This provides a low-level error detection capability, which is combined with a checksum error detection scheme, to yield a dual error detection capability. This, combined with the wider Data Separator capabilities of the TTL design, have provided a very reliable system. This basic design has been in operation for about 5 years, so is not an "unproven" system (I've also been told, but have NOT verified it, that both Commodore and Apple use a GCR encoding system on there Disk Systems).

The TG-99 Disk Interface is the Disk Controller only, which plugs into the Cartridge Slot on the Color Computer. You will need a Disk Drive Interconnect Cable with a 34 Pin Edge Connector to plug into the interface, and you may want to add an external +5VDC Power Supply also. The Power Supply in the Color Computer is marginal, and though it will probably drive this Interface without too much trouble, the TTL Chips will be pushing it. The Controller will support single or double density (tracks), single or double sided drives such as the Shugart 400 series, Siemens 82, TEAC 50 series, Pertec FD200, etc. The TG-99 will support up to 4 Standard 5 1/4 inch Drives, 48, 96, and 100 Tracks/inch.

Another note on the TG-99 Floppy Disk Controller: Dave Allen of TALLGRASS Tech. Corp. has announced that they have signed an agreement with CHAINES ENGINEERING in Shawnee, Kn., licensing them as a "second source" for the Controller. Dan Ditto and his wife (I think she probably does all the work) will be producing the PC boards that will be used for production of the TG-99. They got into the PC Board production business because they could not obtain quality "gold plated" boards (the old "if you want something done right, do it yourself" routine), so the Production Boards will have plated-thru holes and "quality" gold plated connector lands. Their address is:

CHAINES ENGINEERING
12616 W. 62 Terrace
Suite 101
Shawnee, Kn. 66216

The FLEA Monitor System is programmed in position-independent code and resides in a 2732 4Kx8 EPROM. When used normally, by installing it in the ROMpack Slot on the Color Computer, it provides a Disk Control/Mach. Lang. Debugger located at $C000. In this configuration, it is initially entered (Coldstart) at $CAA8, which initializes the 6883 SAM for 4K or 16K RAM chips. FLEA also contains its own Keyboard, Display, and Printer Driver routines, making it completely self-sufficient. TALLGRASS has available the TG-23 64K Col. Comp. Adapter Board, which allows the installation of 64K RAMS such as the Motorola MCM53 or Hatachi HM4864 type chips, providing a 64K memory system. Installing FLEA at the top of RAM and Coldstarting at $xx10, which initializes the SAM for 64K Chips, yields a complete Disk Based Operating System for the Color Computer (Note: mods required for this configuration will void the T.G. 90 day warranty), there is a little unused space in the chip which could be used for a Tape Routine, or the Tape routine could be loaded from Disk, providing a full-function Computer with 48K RAM -- worth thinking about.

When FLEA is installed in a TG-99 in a normal Color Computer configuration, it is a Disk Control system for storage of machine language and BASIC programs, plus a machine-language oriented debugging system. FLEA uses single-letter commands for the operation of the Monitor, Disk Operating, and Monitor systems.

The Monitor Commands allow the user to Enable Screen Display, enable Printer, Kill both (then reenable the one you want), and Jump to an address (for program execution, or whatever).

The Disk Operating System Commands include Initialize diskette, Select Drive, Save file, Load file, Load and Run program, Add file, Change file, and list Directory.

The rest of the commands are primarily Debugging Commands. This system is strictly machine-language oriented, and includes a basic single-step disassemble command, set target address, find target address, fill memory with a byte, set breakpoint, execute program from stack (continue after breakpoint interrupt), compare blocks of mem., display registers, establish software interrupt, input hex to memory, display word and go to next, display word and go back one, display same byte, go back one byte, delete byte, display as alphanumeric, display in IBM format, compute checksum of block of memory, move block of memory, add or subtract words (uses RPN), and convert hex to decimal. It takes a while to get used to all of the commands, but you can look at, change, and control about any way you want.

The simplicity of the commands, especially the Disk operations, make it easy to write programs for this Monitor. As I stated before, it is strictly machine language oriented, which is useful when it gets down to the final program debugging, but is not meant for the average Computer User. TALLGRASS indicates they are not in the program writing business, they make the Hardware. They are looking for independent Software writers to write programs for the system! Bill Vergona of CER-COMP has what appears to be an excellent Disk Operating System for the TG-99 Floppy Disk Interface (also reviewed in this column this month), I'm sure others will appear. Also, for the price, FLEA is handy to have available.

QUICK LOOK:

CER-COMP C-C Mdisk+9

a ROM-based DISK OPERATING SYSTEM (DOS) for the
TALLGRASS CORP. TG-99 Color Computer DISK INTERFACE

(will work with 4K, does NOT require EXT. BASIC --- approx. $70.00)

CER-COMP
3566 Ricochet Ave., Las Vegas, Nv. 89110

C-C Mdisk+9 is a full-functioning DOS ROM which will plug into the ROM socket in the TALLGRASS Corp. TG-99 Color Computer Disk Interface board. Mdisk+9 allows direct Disk control under BASIC operation just like the Tape control commands. Disk Control commands supported are:

LOAD(A) "iN filename" --- Like CLOAD(with ABC)) options). The Drive spec. (iN) defaults to 0 if omitted; all Drive Specs. can be a no. designation OR a 6-char. max Disk Name, such as BASLIB, ADOS9, etc.

SAVE(A) "iN filename" --- Same as the normal SAVE with LOADs' specs.

CHAIN "iN filename" --- A Load and Go command, single command for LOADing a program and then RUNning it.

CDOS "DOS command string" --- If CDOS <ENTER>, transfers control to Mdisk+9; if a "command string" is entered, provides normal DOS commands while remaining in BASIC. When under Mdisk+9 Control, the command 'BASIC' returns you to normal BASIC control.

NOTE: All Disk Command parameters can be passed as a string variable within a BASIC program (2140 LOADA is, for example).

Mdisk+9 provides full File I/O control, also. More 1 BASIC uses the file designations of #-2 for the printer, #-1 for Tape, and #0 for Screen; Mdisk+9 supports #-2 thru #9 for the Disk Files. The normal OPEN/CLOSE routine is used, with a 'D' added to specify Disk, as follows: OPEND "I/O", #N, "filename"; PRINT #N,A$,B$,C$,"TESTING STRINGS"; INPUT #N,A$,B$,D,F ; EOF (N); and CLOSE #N,#L,#1 . LINE INPUT #N,A$,B$ is supported IF you have EXT. BASIC, NO OTHER Mdisk+9 command requires EXT. BASIC. Also included in this group of commands

is the REWIND #N, <sector> command. This automatically CLOSEs a file open for "I" or "O" and reopens it for read at the first sector of the file. If the file was a "R" random access file, the <sector> spec is required.

Operations under Mdisk+9 control provide complete Disk File control and error analysis. Mdisk+9 presently supports 9 file types: including CORES (Ed/Assembler), BASIC Programs, Directory or Data, Text, Multi-task programs, Single-task programs, Pascal Source programs, Pascal P-code programs, and BASIC-Binary Program files. It provides 22 error code descriptions, and supports a File Information Block (FIB) system. It also supports the use of a slash "/" as a "any and all after" character and the question mark "?" character as a "wild card" character for file searches, etc.

The Mdisk+9 Commands are:

LOAD --- Normal load from disk.

(RUN) --- An "implied" command. If a command cannot be found in the command table, it is assumed to be a "Load and Go" command like the CHAIN command in BASIC.

SAVE --- Again, the normal Save to Disk command.

REMOVE --- Removes files from the Directory of specified Disk, removing access to that program or programs.

CHANGE --- Change the Filename.

ANALYZE --- Like Directory in other systems. Analyze the Directory: provides file type, sector count, sector link if a continuation file, beginning memory location, ending memory location, beginning disk sector, and amount of empty space with sector count and loc.

BOTO --- A Gosub command allowing exit of Mdisk+9 to another program other than BASIC.

CHECK --- Check existing files for read errors. Only disturbs Buffer Memory.

NEW --- Initialize Diskette allows up to 38 char labels, with first 8 being the Disk Name.

STRACK --- Sets number of sides and tracks for a specific drive: non-volatile with RESET.

BASIC --- Exit to BASIC.

Mdisk+9 provides a dynamic character file I/O system for external program use and access along with the resident command system just presented. This is the FILE INFORMATION BLOCK System. It allows communication with a disk file on a character at a time basis for use with sequential or random access files, and the FIB automatically allocates sectors as the file requires space. The FIB contains information about the file, such as which drive it is on, next sector to read/write, byte count, file name, file type, first sector of the file, current entry link and sector buffer, etc. Files can be opened for read, write, or update, and can be accessed for random read/write operations. The FIB System is accessed by a JSR to the FISCAL Routine, with a function code in Acc-B and IX pointing to the FIB. Any error sets the Carry Bit in the CC upon return, with an error code in Acc-A.

Even though the Mdisk+9 ROM we recieved for this preliminary report was not yet in final form, the instructions were amazingly complete. Besides the normal Command discussions and a fairly complete description of the FIB System, a full discussion of the user accessable subroutines was included. All entry points are based on the base address of the ROM (#C000 in the TG-99 Disk Interface), with the main entry for initialization of Mdisk+9 being located at the normal Base+#3 (Color BASIC looks for a "OK" at #C000 during RESET to see if there is a Disk Controller in the Computer, and if so, transfers control to #C003 during Power-On RESET).

CER-COMP's Mdisk+9 will provide a "normal" Disk Operating System for the TALLGRASS Corp. TG-99 Disk Interface, making the combination a very cost-effective Disk System for the Color Computer. The Commands are straight-forward and easy to use, and the FIB and user accessable routines allow easy conversion of Color Computer Software to this system. Mdisk+9 DOS in combination with the high storage density of the TALLGRASS Controller provides a strong Disk System at a comparatively low price when combined with the lower priced Disk Drives. Also, Bill Vergona of CER-COMP has the CO-RES9 Assembler operating on Mdisk+9, along with a Disk Editor, Disassembler, and a Disk Utility package. All of this Software should be available about the time this column is published. We'll be reviewing it as soon as we can, so "stay tuned".

REVIEWS:

CER-COMP
5566 Ricochet Ave., Las Vegas, Nv. 89110

CO-RES9 v1.3 --- A Cassette Tape CO-RESIDENT EDITOR/ASSEMBLER - $39.95
(requires 16K ram: does NOT need EXT. BASIC)

CO-RES9 is a 6809 processor each, lang. program formatted for the Color Computer which loads into a little over 9K of memory (which leaves approximately 4K of program useable memory in a 16K machine). The Editor and Assembler are completely integrated, with the Editor being basically the same one reported on in the September 1981 issue of this magazine (and still being used in writing this column). The only Editor Command not included in CO-RES9 is the RENUMBER Command, which allowed adjusting GO-TO's, GOSUB's, etc. when resequencing BASIC Statements with the Color Computer EDITOR. All other Editor Commands are the same, with the additional Assembler Commands added. The Assembler is physically located ahead of the Editor in the program's organization, and four Commands have been added. These are ASMB (enter the Assembler portion of the program), SKIP (the same as the BASIC command SKIPF "FILENAME"), PATCH (executes a SWI2 to call a Monitor), and RUN (like the BASIC command). Therefore, this report will only cover these additions: and to those who missed the EDITOR report, it is a very good Line Editor (oh shucks, call Don and have him send you a copy of that issue, there's a lot more good info in it, too).

The CO-RES9 program contains a full-functioning, non-macro, assembler which supports the standard Motorola Assembler formats. It features listing pagination, most normal Assembler Directives, cross-assembly of 6800 source code, etc. It supports the forcing of extended or direct addressing with the use of the ">" and "<" symbols, but does NOT support the SETDP directive. The listing output will be flagged in the left margin next to the address with a "*" when a non-zero direct page address is forced, to provide a warning. It is suggested that the "<" be used to force Direct Page Addressing: should the "*" show up, you can verify the DP addressing validity. The output listing will also flag an Extended Branch or Jump when they are not required with a ">" symbol in the left margin. CO-RES9 also differs slightly from "standard" in that the auto-decrement format can be specified EITHER as -R, --R OR as R-, R--; both will work OK. Also, the 6809 cross assembly capability allows the use of the clear or and simpler use of INI for LEAX 1,X , CLC for ANDCC #$FE, etc.

ASMB --- This Command enters the Assembly portion of the CO-RES9 program. The Assembler displays "ENTER PASS: 1(P,S); 2/3(T,P,L,M,O)" with a ">" prompt. The 3 passes are: 1, Build Symbol Table; 2, generates output without the Symbol Table being output; and 3, generates the complete output. Pass 1 is used for error analysis in checking the source code: IP builds a NEW Symbol table, and IS adds to the previous table. Pass 2 does not generate a Symbol Table, and therefore requires that Pass 1 or Pass 3 have already been accomplished. It is used in generating a program listing only (2L) or object code only direct to tape (2T). Pass 3 generates a complete Symbol Table and Code output. The 2/3P generates object and listing for the specified options: 2/3T generates object only for the specified options: 2/3L generates a listing only: 2/3O generates a Tape Object files: and 2/3M generates output to memory. Preceding any of the choices with a P activates the Printer: i.e., P3L produces a complete hardcopy listing.

Assembler Directives supported are as follows:

ORG	define new origin (i.e., PC=....)
END	signal the end of the source program
RMB	reserve memory byte
FDB	form double byte
FCC	form constant code
FCB	form constant byte
EQU	assign value to symbol
PAG	skip to top of next page
SPC	skip specified number of lines (SPC4 skips 4 lines)
NAM	specify program name (first 8 chars used - can use 30 total chars to provide additional comments at top of Page
OPT	set or reset assembler options
ATH	define author line comments printed at bottom of page

CO-RES9's outstanding features are it's pagination capability, the use of standard BASIC's ASCII Tape formatting of Source Code information, and it's operation in RAM which allows user modification and adaptability. The Paging option is selected by default, and produces a "prettier" output listing, along with allowing comments in both the NAM and ATH lines for the insertion of dates, versions, notes, etc., with the listing. The standard Tape File use of the ASCII format allows easy program exchange without formatting worries. The Program's residence in RAM will allow mods and patches should requirements change, such as the use of Disk Systems as they become available (the two obvious drawbacks are that it uses up some of the already limited RAM and that the program will get "bombed" quite often). Other features provided with CO-RES9 are that it allows approximately 40 chars of "comment" room, and the output listing also includes the Editor line numbers, allowing easier problem location. Finally, CO-RES9 supports the standard asterisk (*) in column 1 of the Source Code to denote a comment line, allowing 66 character comments.

The instructions supplied with the program discuss Start-up Procedures, an explanation of each of the commands, and several pages of discussion of the Assembler Operation. The Assembler discussion provides information on the Error codes, Assembler Directives and Options, 6800 Cross-assembly, and Position Independent code. This set of instructions did not include the "DEMO" Program, which is a short demonstration program which is extremely helpful in answering detail operational questions and allowing experimentation with a known good program (I had a copy of it on the original release of CO-RES9 V1.0). It seems that some of the suppliers which sell this Software had been leaving them out, but discussions with the author yielded the information that they would be included in shipments shortly, and I think a note to Bill at CER-COMP, with a stamped return envelope, would get you a copy of the listing if you wanted it. In general, the instructions supplied are fairly complete.

This is another good program for the Color Computer from CER-COMP, and is highly recommended. Another AAA Rating to his credit.

REVIEWS:

CER-COMP
5566 Ricochet Ave., Las Vegas, Nv. 89110

TRSMON V1.0 --- A Mach. Lang. Cassette Tape System Monitor --- $19.95
(also available in ROM --- $34.95

TRSMON is a 2K operating system designed for use in any TRS-80C COLOR COMPUTER System. It provides all the standard functions found in most system monitors as well as a Printer/Terminal Driver package. Printer and Terminal modes can be used at rates varying from 300 to 9600 baud, and can be changed at any time. The output can be directed to the printer (RS232 or "SERIAL I/O") port by simply using a "?" preceding a Monitor Command. All output drivers contain a pause feature which allows the output to the screen or printer to be temporarily stopped by hitting any key on the keyboard. TRSMON Commands are two-letter commands followed by their required parameters. The Command input line is buffered and will allow backspace, clear, break, and enter editing during entry to provide error-free command inputs. The Monitor normally expects numeric entries to be hexadecimal.

The Commands supported, and their function, follow:

ME <address) --- Memory examine/change.
Displays the Address followed by it's data byte. The up arrow displays the previous location, any other non-hex entry displays the next location. To change the data, enter a new two-digit hex byte and any non-hex key: a "?" will be displayed if it does not enter correctly (such as trying to write to ROM). <ENTER> exits the mode back to the monitor.

GO <address> --- Go to the specified address for execution, or resume program from the stack PC after breakpoint.

TL <address> --- Tape Load a program (with a specified offset). This loads machine language programs.

TS (begin addr) (end addr) (exec addr) (filename) --- Save machine language program to tape; must contain start and end addresses.

DL <address> <message> --- Down Load Motorola S1-S9 formatted file via the RS232 port. Allows loading the standard tape file format used by most 6800/6809 systems to the Color Computers memory thru the "SERIAL I/O" port. The baud rate can be set with the BR command. The optional <address> specifies an offset for loading, and the <message> causes the Color Computer to send it out (possibly telling the other system you are ready to recieve, etc.). When initiated, the routine waits approx. 20 seconds for "incoming mail", then displays an error message and returns to the monitor. Computer to Computer transfers can be accomplished at up to 9600 baud.

US (begin addr) (end addr) <message> --- Save or Send a file in the S1-S9 format. Basically the same options as DL above.

BR <value> --- Set Baud Rate! 300 is default if BR <ENTER> is keyed. Value is: 0=300, 1=600, 2=1200, 3=2400, 4=4800, and 5=9600 baud.

RS <value> <name> --- Register Set or display. RS <ENTER> displays the registers, entering a hex value and A, X, B, etc. sets that value in that register and then displays the registers.

BS <address> <address> <address> <etc.> --- Set BreakPoints at the specified addresses. Inserts a SWI at the specified location and saves the replaced data. When executed, displays registers and returns data back to the location; GO <ENTER> continues the run.

RB <value> --- Remove Breakpoint. If no value specified, clears the breakpoint table.

DM (begin addr) (end addr) --- Dump Memory in hex and ASCII format. Dump to Screen format is 8 bytes per line with the ASCII code below the hex bytes. ?DM (addr) (addr) provides an excellent printer output of address, 16 hex bytes, and the 16 ASCII codes all on the same line (non-ASCII codes display a period). This provides efficient use of paper with an easily interpreted output.

DA (begin addr) (end addr) --- DisAssemble memory into assembler format. This is not a full-functioning Disassembler, but is designed to be an aid in debugging programs. Output is address, op-code, and operand byte(s). All relative branch instructions also display a '>' followed by the destination address. It does not distinguish between op-code, text char., and data bytes, and is not designed to be fool-proof! Just a convenient aid.

TM <buffer begin> <buffer end> --- Terminal Mode w/ optional Buffer. This function allows the use of the Color Computer as a Video Terminal. The optional buffer is required for any Baud rate over 300 baud; and is not used, even if specified, for 300. The normal buffer is defaulted at 128, more can be allocated. Control is provided to back up display (recieved data will be ignored while in this mode) for review, and to control Echo for full or half Duplex operation. Control chars. can also be sent as required.

BM (begin) (end) (destination) --- Block Move of memory.

FM (begin) (end) (byte) --- Fill Memory with the specified data byte.

FI (begin) (end) (byte) (byte) (byte) (etc.) --- Find byte sequence in memory. Display is the located sequence with the preceding and following bytes displayed on each side of it.

BA --- exit monitor to BASIC.

RP --- exit monitor to ROM Pack at $C000.

IT --- re-initialize the monitor! i.e., Coldstart.

?(Command) --- direct output to the Printer during the execution of the Command.

The instructions supplied with the Tape are more than adequate for this program. They define the use and capabilities of the program, summarize the Commands, specify the ERROR CODES (the program contains 11 two-letter Error Codes - extremely helpful!), and then discusses each Command, providing examples of each. The program user will have very few problems getting this one "up and running".

In summary, this is an excellent System Monitor for the Color Computer, with capabilities not found in most others. It is a welcome addition to the "stable" of Color Computer Software being developed by Bill Vergona of CER-COMP, and is highly recommended to anyone that wants a good Monitor for this machine. Definately a AAA Rating.

BOOK REVIEW:

MICROCOMPUTER ARCHITECTURE and PROGRAMMING
by John F. Wakerly
Copyright 1981 by John Wiley & Sons, Inc.

The Microcomputer explosion of the past few years has also created a "proliferation" (is that a Copyrighted word??) of literature relative to the same subject. There are Books on the Hardware, Software, Systems (both Hard and Soft), Languages, Methods: do's, don'ts, "this is the only way to go", GOTO's are the worst thing on earth (do they even cause damage to the CPU Chip???), etc., etc., etc. But, like Ruby mining, if you sift thru enough rocks, you'll find a jewel, and this is one of the finest. No, this book won't tell you how to spool up your printer by defining your data precisely this way, then build this procedure and that procedure like so (you didn't forget that required space or comma, did you, or, heaven forbid, actually USE that RESERVED word 'there' instead of 'here', did you?), and finally, call the compiler from your remaining 8K of memory and set it if it would consider taking a look at this program and, if it's not too much to ask, would it consider converting this to assembling the 40 pin chip in your computer can digest. It WILL give you a good idea of how to go about it with assembly language, and you will have a good understanding of how to do it with your Microprocessor and how it could be accomplished with other types of Microprocessors. In short, if the "recursive reentrancy of position independent code" leaves you a little dizzy, this book will definately clear the air.

MICROCOMPUTER ARCHITECTURE and PROGRAMMING was written to be used as a text for an introductory course on (microlComputer organization and assembly language programming in a computer science curriculum. It assumes that the reader has a working knowledge of programming principles in a high-level language, and does not attempt to teach "structured programming", as such. It is what we called a "survey" course when I was at college; that is, it gives you the foundation to build on. The book is organized into three parts:

Part 1, the first four chapters, presents introductory material for those of us who haven't just finished a recent programming course. It provides the foundation to be able to follow the rest of the book, and covers term definitions, fundamental concepts, a basic discussion of Pascal so we can follow the algorithms and internal operations of a computer which are presented in the book, a discussion of data structures including arrays, stacks, and queues, and a discussion of the basic concepts of number systems and arithmetic operations used in typical computers.

Part 2, the next eight chapters, is the "heart" of the book, and describes basic principles that a applicable to all computers. This is accomplished by using two "hypothetical" computer processors with differing structures, whose instructions and features are subsets, or parts of, two real processors. The two processors used are the M6800, an accumulator-based processor, using a subset of the Motorola 6809, and the M68000, a register-based processor, using a subset of the Zilog Z8000. This section of the book describes basic instructions, organization, and assembly languages; covers topics on relocation and linking, position-independent code; macros, and structured assembly language; discusses popular addressing modes of different processors, including memory mapping and memory management; explains the format and effects of the most popular computer operation types; examines the high-level concepts of subroutine calling and parameter passing conventions and the concepts of recursion and coroutines; discusses input/output architecture and related software structures; covers interrupts, traps, DMA, and introduces the advanced concepts of processes, shared data structures and reentrancy; and takes a look at software engineering and the program development process.

Part 3, the final seven chapters, examines seven contemporary microprocessors, one per chapter. The first one covered is the PDP-11 and LSI-11, because all contemporary processors "learned" from it. Then the Motorola 68000, Zilog Z8000, TI 9900, Mot. 6809, Intel 8086, and Intel MCS-40 are covered. Each is examined in six sections, which parallel Part 2 of the book: Basic Organization; Assembly Language; Addressing; Operations; Sample Programs; and Input/Output, Interrupts, and Traps. The processor descriptions are fairly detailed, so that the reader can make a fairly accurate evaluation of each processors strengths and weaknesses as they might apply to his requirements. Hardware details are not supplied, nor is all of the details of assembly language, instruction side-effects, development system operation, etc., but the reader does have enough information to understand what would be involved in using each processor in a computer environment.

For the Educators in the audience, Mr. Wakerly also included a "Preface to Instructors", which presents a possible course outline, programming assignments, discusses programming environment, and suggested additional materials.

John Wakerly has done an outstanding job in the organization and presentation of the material in this book. The material is presented in a logical and concise manner, which makes it easy to understand, and the "relaxed" manner in which the book is written is a welcome departure from the normal dry, rigid, formal presentation normally found in a "college textbook". Finally, the "References" at the end of each chapter are a real "goldmine", and make the price of the book worth every penny for that information alone. The presentation of References is also a departure from the norm, in that they are not listed in the formal format, but are introduced and thru discussion, so that the reader has a good idea of the information that the Reference contains (how many times have you wondered if the Reference was used for just one sentence or paragraph, or is the whole thing going to be of value to you).

MICROCOMPUTER ARCHITECTURE and PROGRAMMING is an outstanding book that anyone interested in learning assembly language programming will have no trouble understanding, and will actually "enjoy" reading. It is error-free, and will provide a welcome and often-used reference for the beginner and "pro" alike. Don't worry about spending your hard-earned dollar on this book, it's worth every penny of it's cost.

RUMORS:

Any of you who have tried the Radio Shack Videotex program on the Color Computer are well aware of how frustrating it is to spend your free hour wondering around all that information on CompuServe like struggling thru a maze, and then realizing how confused you are about everything you read and what it meant. Don't get me wrong, CompuServe is FANTASTIC! I used the first hour and don't think I even got started finding out what they have, let alone how to use it effectively. WOW!! Or hooked up to one of the many other systems, such as Star-Kits, etc., and try to figure out the "file-page structure" provided for 80-column Video Displays on the 32-column Display on the Color Computer -- GOOD LUCK. If Peter Stark hadn't come on line (if you haven't had the Display start answering you in sort of a "stilted" manner, with precise details, you're in for another surprise - I've heard of ELIZA, but it's not THAT good; thanks for the rescue, Pete), I'd still be trying to figure that "file-spec" out. I'll say this, that first night with Vedlotex was DIFFERENT. If you haven't tried Radex/Videotex type Computer Operations, grab one up first chance you get. "There GOLD in them ther hills!" Seriously, it's a whole new Ball Game, and more and more groups are setting up systems. The 2 to 3 hours on the Modem that night cost me about $25.00 (with the 1 free hour on CompuServe, which is only $5 to $7.00/hour normally).

Anyway, the BIG problem with the Radio Shack Vedlotex system is that all you can do is watch the Screen, and all that information just "disappears" (with some work, you might be able to save what's

LEFT in memory to Tape, but the program does NOT provide a way). Bob Lentz and crew at The MICRO WORKS is working on a system that will be worth every penny it might cost if they get enough requests for it. It will probably be along the lines of a ROM PACK for the Cartridge Slot on the Color Computer with a pig-tail "SERIAL I/O" type connector for a Printer hook-up. (The Modem plugs into the Color Computers "SERIAL I/O" output, so there's no way to hook up a Printer with the Modem in). The program would be in ROM, leaving all of memory free, and control capabilities for switching the Printer on and off. This would allow making a "hard-copy" of anything interesting you see. PLUS provide "readable" 80-column information. Drop them a line for info or to show support of the idea; personally, after the above experience, I would call it a MUST if we are going to have a "useable" system on the Color Computer.

"LOG OFF"

Well, another "chunk" of the magazine eaten up - much more of this and Dan will have to add more pages to the magazine. We've been concentrating on "reviews" these first several columns, and will continue to do so, because of all the products that are literally "exploding" onto the market for the Color Computer. We feel that you need to know what these programs offer because they will be forming the "basis" of your Computer System for the years to come - the Disk Systems, Assemblers, Monitors, etc., many of which represent a major cash outlay. The future of the Color Computer is extremely bright; the next several months will see more and more products becoming available, and we'll have to draw a line somewhere as to what we can cover in the detail I have been covering, and what we'll have to summarize (this isn't a 500 page magazine, yet). Let us know what you want to see, and we'll go that way. Hopefully, we can start looking at the "innards" of this machine some shortly, but I have felt it was more important to present a detailed look at most of the products we have reviewed these first few columns --- and from the letters I've recieved so far, we seem to be pretty much "on target" (actually, I'm astounded, and a little scared, at the response I've gotten so far - the first "Users Notes" just hit the Mailboxes less that a week ago). Drop us a line, but PLEASE include a self-addressed, STAMPED, Envelope if you would like a reply, and I'll try to at least point you in the right direction. To those that have written, THANKS for the comments; we'll see you next month.

SIMULATION, GAMES, AND RANDOM VARIABLES

by
T. F. Elbert
University of West Florida
Pensacola, Florida 32504

PART 2

These principles can be exemplified by exercising the random integer generator of Listing 1 with the following values for the parameters:

$$M = 128;\ A = 5;\ C = 3;\ X_0 = 100.$$

These values of M, A, and C satisfy the three conditions for maximum period (128), and indeed the computational results will verify this. The reader should also note that zero is a possible value in this case, and will occur in any sequence generated. On the other hand, the values

$$M = 127;\ A = 5;\ C = 3,\ X_0 = 100$$

produce a sequence with period equal to 42. Choosing some other value of X_0 may produce a different sequence, depending upon whether or not the value 100 appears in the original sequence. For example, $X_0 = 52$ produces the same sequence as above, while $X_0 = 50$ produces a different sequence with the same period. The values

$$M = 127;\ A = 8;\ C = 3;\ X_0 = 100$$

produces a sequence with period length of 126, one less than the maximum possible length. A sequence such as this is just as useful as one with maximum length.

Further insight into linear congruential sequences can be obtained by recognizing that, if A and M are not relatively prime, and if

$$d = \gcd(A,M)$$

then the sequence can produce only M/d distinct values. Thus, a loop of period M/d will be entered after the first calculation. This can be readily exemplified by choosing the values

$$M = 143;\ A = 11;\ C = 3;\ X_0 = 100,$$

in which case there are only

$$d = \gcd(11,143) = 13$$

possible values. The result is the loop

102, 124, 80, 25, 135, 58, 69, 47, 91, 3, 36 113, 100

which is entered at the first iteration. Even more serious is the situation where M is a power of two, for in this case, when A is even the sequence eventually enters a loop of period 1. Such an instance is illustrated by the values

$$M = 128;\ A = 4;\ C = 3;\ X_0 = 100.$$

After four iterations, the sequence is constant at X = 127.

The behavior of the linear congruential sequence is described by the relationship

$$X_{n+k} = (A^k X_n + (A^k - 1)C/(A-1)) \bmod M$$

from which the (n+k)th term can be determined directly from the n-th term.

When n=0 in this expression

$$X_k = (A^k X_0 + (A^k-1)C/(A-1)) \bmod M$$

This gives the k-th value as a function of the seed value X_0. When M is a power of two and A is even,

$$M = 2^e \qquad A = 2^p q$$

where q is an odd integer, then

$$X_k = (2^{pk} q^k X_0 + (A^k-1)C/(A-1)) \bmod 2^e$$

By using the basic identity

$$ab \bmod M = (a \bmod M)(b \bmod M) \bmod M,$$

this can be written as

$$X_k = ((2^{pk} q^k X_0) \bmod 2^e + ((A^k-1)C/(A-1)) \bmod 2^e) \bmod 2^e.$$

But, for $pk \geq e$,

$$2^{pk} q^k X_0 \bmod 2^e = 0$$

so that the value of X_k is

$$X_k = ((A^k-1)C/(A-1)) \bmod 2^e$$

independent of X_0. Furthermore, for $pk \geq e$, this expression is also independent of k, by virtue of the same argument presented above. This result implies that the sequence reaches a constant value within K steps, where K is the largest integer greater than e/p, regardless of the value of X_0. This behavior has already been noted in a previous example. To illustrate this concept further, the values

$$M = 128;\ A = 6;\ C = 3;\ X_0 = 100,$$

yield K=7 and for k =7,

$$X_7 = ((6^7-1)\ 3/5) \bmod 128 = 25$$

That this sequence is constant at 25 after 7 steps can be readily verified by use of the computer program of Listing 1.

An important special case arises when C=0, in which event the sequence is given by

$$X_{n+1} = AX_n \bmod M$$

This form is more convenient from a computational viewpoint, but has the potential drawback that, if M is divisible by any integer q, and X_0 is a multiple of q, then all subsequent values of X will also be multiples of q. This fact can be exemplified by setting

$$M = 143;\ A = 7;\ C = 0;\ X_0 = 39.$$

The resulting sequence has a period length 10, and contains only multiples of 13. Again, verification is possible by use of the program of Listing 1.

To avoid a situation such as that just discussed, a random integer generation sequence with C=0 must have X_0 and M relatively prime for all n. This means that all integers X for which

$$\gcd(X,M) = 1$$

must systematically be excluded from the sequence, thus shortening the maximum period possible with the sequence. This is the price paid for computational convenience, but it may not be too severe. For example, if M is itself a prime integer, then the period can be of length (M-1), only one less than the maximum possible length. A previous example, with M=127, has illustrated this concept.

There is a general rule concerning the period of a linear congruential sequence with C=0 (Reference 1), which states that the maximum period is achieved if

B1) X_0 is relatively prime to M.

B2) A is a primitive element modulo M.

The first condition is apparent from the previous discussion. The second concerns a mathematical entity called a "primitive element" and, while a full discussion is beyond the scope of present considerations, certain special cases are easily considered. An integer A is a primitive element modulo M if:

C1) $m = 2^e$, $e \geq 4$, and A mod 8 = 3 or 5.

or

C2) M is odd, A mod M = 0, and $A^{(M-1)/q} \bmod M = 1$, where q is any prime divisor of M-1.

The first condition considers the important case when M is a power of 2, while the second applies to, among others, the

case where $M=2^e \pm 1$. Reference 1 contains information on how to determine multiplier A for any general value of M.

The value of the maximum period length can be determined in the following manner, and is function only of M. The modulus M can be expressed in terms of its prime factors

$$M = (p_1^{e_1})(P_2^{e_2})\ldots(p_k^{e_k})$$

in which case the functions

$$f(p^e) = \begin{cases} (p-1)p^{e-1} & \text{for } p \neq 2 \\ 2^{e-2} & \text{for } p = 2 \end{cases}$$

are defined. The length of the maximum period is then the least common multiple of the $f(p^e)$. The least common multiple is the smallest integer into which each of the $f(p^e)$ values will evenly divide. When M is a prime number there is a single prime factor p, given by

$$p = M$$

which results in

$$f(p) = p-1 = M-1$$

and the maximum period is (M-1), as previously noted.

To illustrate these concepts, the values

$$M = 127;\ A = ?;\ C = 0;\ X_0 = 100$$

can be used. Since 127 is prime, the maximum period is of length 126. Also, since

$$(M-1) = 126 = (2)(3^3)(7)$$

condition (C2) above requires that the value of A, in order to achieve this maximum period length, must satisfy the conditions

$$A^{63} \mod 127 = 1$$

$$A^{42} \mod 127 = 1$$

$$A^{18} \mod 127 = 1.$$

These tests cannot be performed directly because of the large numbers encountered, but the modular exponentiation technique discussed in Reference 2 can be used. A PASCAL program implementing this technique is given in Listing 3. By use of this program it can be determined that A=7 satisfies the requirement above, while A=11 does not. The reader may use the program of Listing 1 to verify that A=7 leads to a period of 126, while A=11 produces a period of 63.

As a second example, consider the values

$$M = 128;\ A = 11;\ C = 0;\ X_0 = 101.$$

Here, condition (C1) applies, and since M is a power of 2, and 11 mod 8 = 3, the conditions for maximum period length are obtained. Furthermore, the value of the maximum length is obtained by recognizing that

$$M = p^e = 2^7$$

so that

$$f(2^7) = (2^5) = 32$$

is the period length. This can be verified by use of the computer program of Listing 1.

When C=0, M is a power of 10

$$M = 10^e,$$

$e \geq 5$, and X_0 is not a multiple of 2 or 5, the maximum period length is

$$5 \times 10^{e-2}$$

and is obtained when A mod 200 has one of the following thirty-two values.

3, 11, 13, 19, 21, 27, 29, 37, 53, 59, 61, 67, 69, 77, 83, 109, 117, 117, 123, 131, 133, 139, 141, 147, 163, 171, 173, 179, 181, 197, 189, 197.

The reader is referred to Reference 1 for further details.

The computation of modular values can be greatly simplified when the modulus M is taken to be the word length of the computer used to generate the sequence. When a high level language is used, integer arithmetic is done with a modulus one greater than the largest integer repre-

sentable in a single precision computer word. Typically, mini- and microcomputer high level languages use 16-bit two's complement integer representation, so that the largest positive integer is 32767, and integer arithmetic is done modulo 32768. If integer overflow does not cause a data exception, then the overflow feature of the machine can be used to accomplish the modular arithmetic. This requires that the modulus M be 32768, and the maximum period length is easily determined to be

$$2^{13} = 8192$$

Furthermore, in order to obtain this period length, the value of A must satisfy condition (B2) above. Since

$$171 \bmod 8 = 3$$

the integer 171 is a primitive element modulo 32768 because it satisfies condition (C1). Thus the sequence

$$X_{n+1} = 171\ X_n \bmod 32768$$

should produce the maximum period length of 8192 if X_0 is odd, and therefore relatively prime to 32768. Listing 4 contains a PASCAL program for generating random integers with a modulus of 32768, using the inherent modular arithmetic provided by integer overflow. With some patience, the reader can confirm the period of the above sequence by use of this program.

To be continued...

DISEMBLE.CMD

This program was written because none of the published disassembler programs that I had tried were satisfactory: either they failed to run at all (possibly due to my already existing code) or else the output was not particularly useful.

When I decided to produce my own program, I thought that ideally it should produce an output as much like an assembly listing as possible. I had already adapted DUMPCMDS.CMD by R.L. Pigford to run under FLEX2, and was so impressed that I decided to make it the first part of my program.

Part two of the program takes over where DUMPCMDS leaves off: it prints the opcode mnemonic followed by the operand. The latter is in the form in which it would usually appear in an assembly listing except that no labels are used.

My usual method of use is:

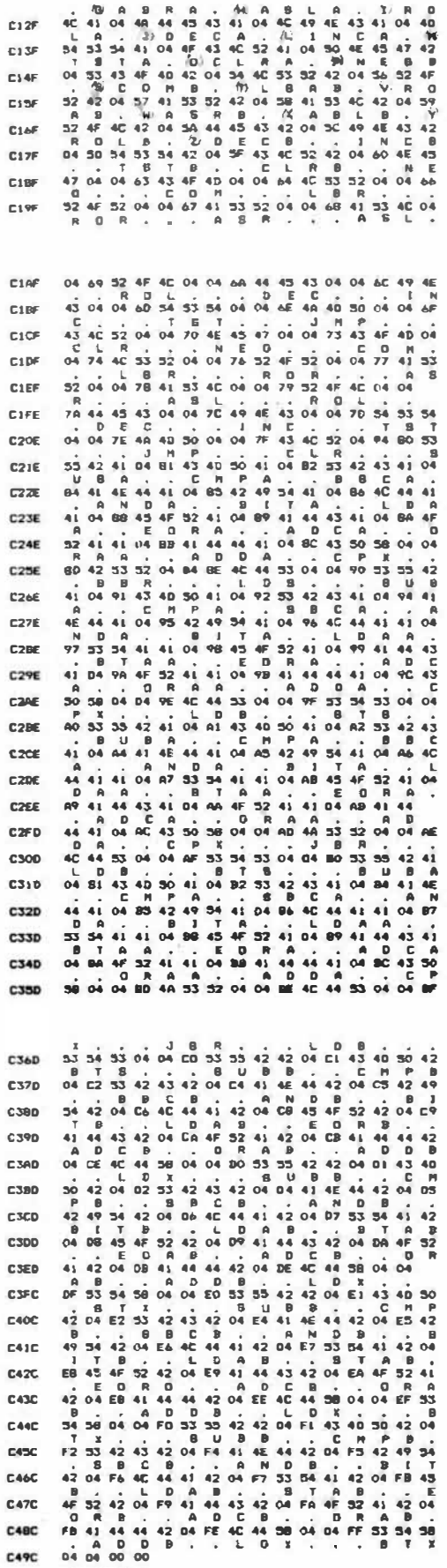
1. Prepare an output file by using the 'O' command.
e.g. O,1.LISTING.OUT,O.DISEMBLE,1.PROGRAM.BIN
2. This can then be edited in the usual way.
a) The first editor command is usually:
C/ / /!
This changes the spacing to allow enough space for labels to be added.
b) Then,
^C/ E/LE/! i^C/ 8/16/!
This allows a tab character to be inserted, so that a neat layout can be achieved.
c) Finally, labels can be inserted and references to memory locations can be changed throughout the listing to the corresponding label.

The next desirable feature would be to strip away the columns of code to allow re-assembly. This I have not done. I think it would be easier to change the Assembler program where the number of bytes to be ignored is specified.
The method used for searching the mnemonic table was for ease of writing, and could probably be improved to allow a smaller table. However, as DISEMBLE, like DUMPCMDS, disassembles from disk without having to have the program being disassembled in memory, the amount of memory needed is not a problem.
Any improvements to the program would be welcomed by me, as also any report of bugs found.

```
DISEMBLE.CMD                    TSC MNEMONIC ASSEMBLER    PAGE 1

                    * FIRST PART IS DUMPCMDS BY R.L. PIGFORD *
                    * MODIFIED BY W.A. HUGHES FOR FLEX2, WITH ADDED *
                    * PORTION TO LIST ASSEMBLER-STYLE MNEMONICS *
                    *

                    * SYSTEM EQUATES
                    *
AD03                WARMS   EQU   $AD03
AD18                PUTCHR  EQU   $AD18
AD1E                PSTRNG  EQU   $AD1E
AD24                PCRLF   EQU   $AD24
AD2D                GETFIL  EQU   $AD2D
AD36                ADDBX   EQU   $AD36
AD39                OUTDEC  EQU   $AD39
AD3C                OUTHEX  EQU   $AD3C
AD3F                RPTERR  EQU   $AD3F
AD45                OUTADR  EQU   $AD45
B403                FMSCLS  EQU   $B403
B406                FMS     EQU   $B406
AB40                FCB     EQU   $AB40
AC1E                TADDR   EQU   $AC1E
C000                TABLE   EQU   $C000     OR ANY CONVENIENT LOCATION
C49E                TABEND  EQU   $C49E
                    *
A100                        ORG   $A100     ASSEMBLE IN USER COMMAND SPACE
A100 20 0E          START   BRA   INIT
A102 02             VN      FCB   2         VERSION #2
A103                SAVEX   RMB   2         COMMAND ADDR. COUNTER
A105                BCNTR   RMB   1         NO. OF BYTES COUNTER
A106                SAVEB   RMB   1         NO. OF BYTES IN COMMAND
A107                COUNT   RMB   1         NO. OF BYTES (USED IN PART 2)
A108                FLAG1   RMB   1         INDICATES INCOMPLETE 2-BYTE CMD.
A109                FLAG2   RMB   1         INDICATES INCOMPLETE 3-BYTE CMD.
A10A                BYTE1   RMB   1         OP-CODE OF COMMAND
A10B                BYTE2   RMB   1         FIRST OPERAND BYTE
A10C                BYTE3   RMB   1         SECOND OPERAND BYTE
A10D                WORKSP  RMB   2         TEMPORARY STORAGE
A10F                STEMP   RMB   1
A110 BD AD 24       INIT    JSR   PCRLF
A113 BD A2 2D               JSR   OPENFI    GET FILE SPEC & OPEN FOR READ
A116 8D 02                  BSR   LOAD2
A118 20 F6                  BRA   INIT
A11A 7F A1 08       LOAD2   CLR   FLAG1     CLEAR FLAGS TO BEGIN COMMAND
A11D 7F A1 09               CLR   FLAG2
A120 BD A2 41       LOADER  JSR   FMS1      GET BYTE FROM FILE
A123 81 02                  CMP A #2        IS IT NEW RECORD INDICATOR?
A125 27 1E                  BEQ   LDR2

A127 81 16                  CMP A #$16      IS IT TRANSFER ADDR. INDICATOR
A129 26 F5                  BNE   LOADER    RETURN FOR ANOTHER BYTE
A12B BD A2 41               JSR   FMS1      GET PROGRAM ENTRY ADDR.
A12E B7 AC 1E               STA A TADDR     & STORE IT
A131 BD A2 41               JSR   FMS1
A134 B7 AC 1F               STA A TADDR+1
A137 CE A2 1A               LDX   #TRN3
A13A BD AD 1E               JSR   PSTRNG    PRINT BEGIN-ADDR MESSAGE
A13D CE AC 1E               LDX   #TADDR
A140 BD A1 F8               JSR   OUT4HS    OUTPUT TRANSFER ADDRESS
A143 20 09          LOAD1   BRA   LOADER    GET ANOTHER BYTE
A145 BD A2 41       LDR2    JSR   FMS1      GET 2-BYTE BEGIN ADDR.
A148 B7 A1 03               STA A SAVEX
A14B BD A2 41               JSR   FMS1
A14E B7 A1 04               STA A SAVEX+1
A151 BD A2 41       LDR5    JSR   FMS1      GET NO. OF BYTES IN RECORD BLOCK
A154 B7 A1 05               STA A BCNTR
A157 27 C7                  BEQ   LOADER    RETURN IF BLOCK IS BLANK
A159 BD A2 41       GPBYLON JSR   FMS1      GET DATA BLOCK FROM FILE
A15C 7A A1 05               DEC   BCNTR
A15F 7D A1 08               TST   FLAG1     2-BYTE CMD?
A162 26 57                  BNE   GAMMA1
A164 7D A1 09               TST   FLAG2     3-BYTE CMD?
A167 26 77                  BNE   DELTA1
A169 B7 A1 0A               STA A BYTE1     STORE BYTE AS OP-CODE
A16C 5F                     CLR B           CLEAR NO. BYTES IN CMD.
A16D CE A1 03               LDX   #SAVEX
A170 BD A1 F8               JSR   OUT4HS    OUTPUT ADDR. OF COMMAND
A173 BD A2 0C               JSR   OUTS
A176 CE A1 0A               LDX   #BYTE1
A179 BD A1 FC               JSR   OUT2HS    OUTPUT OP-CODE
A17C BD A2 12               JSR   INCSAV
A17F B6 A1 0A               LDA A BYTE1     GET OP-CODE AGAIN
A182 81 8C          INSTYP  CMP A #$8C      TEST FOR NO. BYTES
A184 27 18                  BEQ   THREEB            IN COMMAND
A186 81 8E                  CMP A #$8E
A188 27 14                  BEQ   THREEB
A18A 81 CE                  CMP A #$CE
A18C 27 10                  BEQ   THREEB
A18E 84 F0                  AND A #$F0
A190 81 20                  CMP A #$20
A192 27 08                  BEQ   TWOB
A194 81 60                  CMP A #$60
A196 25 08                  BCS   ONEB
A198 84 30                  AND A #$30
A19A 81 30                  CMP A #$30
A19C 26 01                  BNE   TWOB
A19E 5C             THREEB  INC B   B REG CONTAINS NO. OF BYTES
A19F 5C             TWOB    INC B           IN THE COMMAND
A1A0 F7 A1 06       ONEB    STA B SAVEB     STORE NO. CMD. BYTES IN SAVEB
```

```
A1A3 F7 A1 07            STA B  COUNT          & IN COUNT
A1A6 26 09               BNE    MU             BRANCH IF 1- OR 2-BYTE CMD.
A1A8 BD A2 6F  END1      JSR    PART2 (JSR PCRLF, IF PART2  NOT REQUIRED)
A1AB 6D 55               TST    TESTB          ANY BYTES LEFT IN RECORD?
A1AD 25 94               BCS    LOAD1
A1AF 20 AB               BRA    EPSILON        GET NEW DATA
A1B1 BD 4F     MU        JSR    TESTB          ANY BYTES LEFT IN BLOCK?
A1B3 24 0B               BCC    GAMMA
A1B5 7C A1 08            INC    FLAG1          SET FLAG TO SHOW INCOMPLETE
A1B8 7E A1 20            JMP    LOADER           2-BYTE CMD;  GET DATA FROM
A1BB 7F A1 08  GAMMA1    CLR    FLAG1            NEXT BLOCK;  CLEAR FLAG UPON
A1BE 20 03               BRA    GAMMA2           RETURN
A1C0 BD A2 41  GAMMA     JSR    FMS1           GET NEW BYTE FROM FILE
A1C3 B7 A1 0B  GAMMA2    STA A  BYTE2          STORE IT AS FIRST OPERAND BYTE
A1C6 7A A1 05            DEC    BCNTR
A1C9 CE A1 0B            LDX    #BYTE2
A1CC BD A1 FC            JSR    OUT2HS         OUTPUT FIRST OPERAND
A1CF BD 41               BSR    INCBAV
A1D1 7A A1 06            DEC    SAVEB
A1D4 27 D2               BEQ    END1           RETURN IF COMMAND INCOMPLETE
A1D6 BD 2A               BSR    TESTB          ANY BYTES LEFT IN BLOCK?
A1D8 24 0B               BCC    DELTA
A1DA 7C A1 09            INC    FLAG2          SET FLAG TO SHOW NEED BYTE 3
A1DD 7E A1 20            JMP    LOADER         RETURN FOR NEW RECORD BLOCK
A1E0 7F A1 09  DELTA1    CLR    FLAG2
A1E3 20 03               BRA    DELTA2
A1E5 BD A2 41  DELTA     JSR    FMS1           GET SECOND OPERAND BYTE
A1E8 B7 A1 0C  DELTA2    STA A  BYTE3
A1EB 7A A1 05            DEC    BCNTR
A1EE CE A1 0C            LDX    #BYTE3
A1F1 BD 09               BSR    OUT2HS         OUTPUT SECOND OPERAND
A1F3 BD 1D               BSR    INCSAV
A1F5 7E A1 A8            JMP    END1
*
A1F8 BD AD 3C  OUT4HS    JSR    OUTHEX         OUTPUT 2 HEX BYTES
A1FB 08                  INX
A1FC BD AD 3C  OUT2HS    JSR    OUTHEX         OUTPUT 1 HEX BYTE
A1FF BD 09               BSR    OUTB           OUTPUT SPACE
A201 39                  RTS
*
A202 7D A1 05  TESTB     TST    BCNTR          TEST NO. BYTES LEFT IN BLOCK
A205 27 03               BEQ    SETC
A207 0C                  CLC                   CLEAR CARRY IF BYTES ARE LEFT
A208 20 01               BRA    RETN
A20A 00        SETC      SEC                   SET CARRY IF ALL BYTES USED
A20B 39        RETN      RTS
*
A20C 86 20     OUTB      LDAA   #$20           OUTPUT SPACE
A20E BD AD 18            JSR    PUTCHR
A211 39                  RTS
*
A212 FE A1 03  INCBAV    LDX    SAVEX          INCREMENT ADDR. COUNTER
A215 08                  INX
A216 FF A1 03            STX    SAVEX
A219 39                  RTS
*
A21A 53        MSG3      FCC    'STARTING ADDRESS ='
A21B 54 41
A21D 52 54
A21F 49 4E
A221 47 20
A223 41 44
A225 44 52
A227 45 53
A229 53 20
A22B 3D
A22C 04                  FCB    4
*
A22D CE AB 40  OPEN1     LDX    #FCB
A230 BD AD 2D            JSR    GETFIL
A233 25 27               BCS    ERROR
A235 86 01               LDA A  #1             FMS CODE FOR READ OPEN
A237 A7 00               STA A  0,X
A239 BD A2 41            JSR    FMS1
A23C 86 FF               LDAA   #$FF           DISABLE SPACE COMPRESSION
A23E A7 3B               STA A  59,X
A240 39                  RTS
A241 CE A8 40  FMS1      LDX    #FCB
A244 BD B4 06            JSR    FMS
A247 26 01               BNE    FMS2
A249 39                  RTS
A24A A6 01     FMS2      LDA A  1,X
A24C B1 08               CMP A  #8             EOF?
A24E 26 0C               BNE    ERROR
A250 86 04               LDA A  #4
A252 A7 00               STA A  0,X
A254 BD B4 03            JSR    FMSCLR
A257 26 03               BNE    ERROR
A259 7E AD 03            JMP    WARMS
A25C BD AD 3F  ERROR     JSR    APTERR
A25F BD B4 03            JSR    FMSCLR
A262 7E AD 03            JMP    WARMS
*
A265 FE A1 03  END4      LDX    SAVEX
A268 7F A1 07            CLR    COUNT
A26B BD A0 24            JSR    PCRLF
A26E 39                  RTS
A26F BD A2 0C  PART2     JSR    OUTB
A272 B6 A1 0A  MNEM      LDA A  BYTE1          GET OP-CODE AGAIN
A275 CE C0 00            LDX    #TABLE
A278 A1 00     CK1       CMP A  0,X            SEARCH MNEMONIC TABLE
A27A 27 0D               BEQ    HIT            BRANCH IF FOUND
A27C 08                  INX
A27D 08                  INX
A27E 08                  INX                   ADVANCE POINTER
A27F 08                  INX                      TO NEXT MNEMONIC
A280 08                  INX                        IN TABLE
A281 08                  INX
A282 8C C4 9E            CPX    #TABEND
A285 26 F1               BNE    CK1
A287 20 49     MSG      BRA    INVALID        TREAT AS ASCII CODE
A289 08        HIT       INX
A28A A6 00               LDA A  0,X
A28C BD A3 52            JSR    PDATA1         OUTPUT MNEMONIC
A28F FE A1 03            LDX    SAVEX
A292 BD A2 0C  DONE      JSR    OUTB
A295 27 CE               BEQ    END4           END OF INSTRUCTION
A297 B6 A1 0A  MODE      LDA A  BYTE1
A29A 81 8D               CMP A  #$8D           FIND EXCEPTIONS
A29C 27 08               BEQ    REL2
A29E 81 CE               CMP A  #$CE
A2A0 27 38               BEQ    IMMED
A2A2 84 F0               AND A  #$F0           ANALYSE INSTUCTION TYPE
A2A4 81 20               CMP A  #$20
A2A6 27 79               BEQ    REL
A2A8 81 60               CMP A  #$60
A2AA 27 59               BEQ    INDEX
A2AC 81 70               CMP A  #$70
A2AE 27 53               BEQ    EXTDIR         EXTENDED
A2B0 81 80               CMP A  #$80
A2B2 27 26               BEQ    IMMED
A2B4 81 90               CMP A  #$90
A2B6 27 4B               BEQ    EXTDIR         DIRECT
A2B8 81 A0               CMP A  #$A0
A2BA 27 49               BEQ    INDEX
A2BC 81 B0               CMP A  #$B0
A2BE 27 43               BEQ    EXTDIR         EXTENDED
A2C0 81 C0               CMP A  #$C0
A2C2 27 16               BEQ    IMMED
A2C4 81 D0               CMP A  #$D0
A2C6 27 3B               BEQ    EXTDIR         DIRECT
A2C8 81 E0               CMP A  #$E0
A2CA 27 39               BEQ    INDEX
A2CC 81 F0               CMP A  #$F0
A2CE 27 33               BEQ    EXTDIR         EXTENDED
A2D0 20 93               BRA    END4
A2D2 86 A1 0A  INVALID   LDA A  BYTE1
A2D5 BD AD 18            JSR    PUTCHR         OUTPUT AS ASCII
A2D8 20 8B               BRA    END4
A2DA 86 23     IMMED     LDA A  #'#
A2DC BD AD 18            JSR    PUTCHR
A2DF 86 24     IMM2      LDA A  #'$
A2E1 BD AD 18            JSR    PUTCHR
A2E4 B6 A1 0B            LDA A  BYTE2          GET OPERAND BYTE
A2E7 7A A1 07            DEC    COUNT
A2EA B7 A1 0D  IMM3      STA A  WORKSP
A2ED CE A1 0D            LDX    #WORKSP
A2F0 BD AD 3C            JSR    OUTHEX         OUPUT OPERAND BYTE
A2F3 7D A1 07            TST    COUNT          END OF INSTRUCTION?
A2F6 27 0B               BEQ    END1MM
A2F8 7A A1 07            DEC    COUNT
A2FB B6 A1 0C            LDA A  BYTE3          GET SECOND OPERAND BYTE
A2FE 20 EA               BRA    IMM3
A300 7E A2 65  END1MM    JMP    END4
A303 20 DA     EXTDIR    BRA    IMM2           OUTPUT A '$'
A305 7F A1 0D  INDEX     CLR    WORKSP
A308 B6 A1 0B            LDA A  BYTE2
A30B B7 A1 0E            STA A  WORKSP+1
A30E CE A1 0D            LDX    #WORKSP        OUTPUT OPERAND
A311 BD AD 39            JSR    OUTDEC           AS DECIMAL
A314 86 2C               LDA A  #',
A316 BD AD 18            JSR    PUTCHR         OUTPUT A COMMA
A319 86 58               LDA A  #'X
A31B BD AD 18            JSR    PUTCHR         OUTPUT 'X'
A31E 7E A2 65            JMP    END4
A321 F7 A1 0F  REL       STA B  BTEMP
A324 F6 A1 0B            LDA B  BYTE2
A327 FE A1 03            LDX    SAVEX
A32A BD AD 36            JSR    ADDBX          CALCULATE RELATIVE ADDRESS & O/P IT
A32D FF A1 0D            STX    WORKSP
A330 F6 A1 0B            LDA B  BYTE2
A333 C1 7F               CMP B  #$7F
A335 23 03               BLS    REL4           ADJUST FOR BACKWARD BRANCH
A337 7A A1 0D            DEC    WORKSP
A33A CE A1 0D  REL4      LDX    #WORKSP
A33D 86 24               LDA A  #'$
A33F BD AD 18            JSR    PUTCHR
A342 BD AD 45            JSR    OUTADA
A345 F6 A1 0F            LDA B  BTEMP
A348 FE A1 03            LDX    SAVEX
A34B 7E A2 65            JMP    END4
A34E BD AD 18  PDATA2    JSR    PUTCHR
A351 08                  INX
A352 A6 00     PDATA1    LDA A  0,X
A354 81 04               CMP A  #4
A356 26 F6               BNE    PDATA2
A358 39                  RTS
                         END

      NO ERROR(S) DETECTED
```

```
C000  01 4E 4F 50 04 04 06 54 41 50 04 04 07 54 50 41
        .  N  O  P  .  .  .  T  A  P  .  .  .  T  P  A
C010  04 04 08 49 4E 58 04 04 09 44 45 58 04 04 0A 43
        .  .  .  I  N  X  .  .  .  D  E  X  .  .  .  C
C020  4C 56 04 04 0B 53 45 56 04 04 0C 43 4C 43 04 04
        L  V  .  .  .  S  E  V  .  .  .  C  L  C  .  .
C030  0D 53 45 43 04 04 0E 43 4C 49 04 04 0F 53 45 49
        .  S  E  C  .  .  .  C  L  I  .  .  .  S  E  I
C040  04 04 10 53 42 41 04 04 11 43 42 41 04 04 16 54
        .  .  .  S  B  A  .  .  .  C  B  A  .  .  .  T
C050  41 42 04 04 17 54 42 41 04 04 19 44 41 41 04 04
        A  B  .  .  .  T  B  A  .  .  .  D  A  A  .  .
C060  18 41 42 41 04 04 20 42 52 41 04 04 22 42 48 49
        .  A  B  A  .  .  .  B  R  A  .  .  (-) B  H  I
C070  04 04 23 42 4C 53 04 04 24 42 43 43 04 04 25 42
        .  .  (#) B  L  S  .  .  ($) B  C  C  .  .  (%) B
C080  43 53 04 04 26 42 4E 45 04 04 27 42 45 51 04 04
        C  S  .  .  (&) B  N  E  .  .  (') B  E  Q  .  .
C090  28 42 56 43 04 04 29 42 56 53 04 04 2A 42 50 4C
        (() B  V  C  .  .  ()) B  V  S  .  .  (*) B  P  L
C0A0  04 04 2B 42 4D 49 04 04 2C 42 47 49 04 04 2D 42
        .  .  (+) B  M  I  .  .  (,) B  G  E  .  .  (-) B
C0B0  4C 54 04 04 2E 42 47 54 04 04 2F 42 4C 45 04 04
        L  T  .  .  (.) B  G  T  .  .  (/) B  L  E  .  .
C0C0  30 54 53 58 04 04 31 49 4E 53 04 04 32 50 55 4C
        (0) T  S  X  .  .  (1) I  N  S  .  .  (2) P  U  L
C0D0  41 04 33 50 55 4C 42 04 34 44 49 53 04 04 35 54
        A  .  (3) P  U  L  B  .  (4) D  I  S  .  .  (5) T
C0E0  58 53 04 04 36 50 53 48 41 04 37 50 53 48 42 04
        X  S  .  .  (6) P  S  H  A  .  (7) P  S  H  B  .
C0F0  39 52 54 53 04 04 3B 52 54 49 04 04 3E 57 41
        (9) R  T  S  .  .  (;) R  T  I  .  .  (>) W  A
C0FF  49 04 04 3F 53 57 49 04 04 40 4E 45 47 41 04 43
        I  .  .  (?) S  W  I  .  .  (@) N  E  G  A  .  (C)
C10F  43 4F 4D 41 04 44 4C 53 52 41 04 46 52 4F 52 41
        C  O  M  A  .  (D) L  S  R  A  .  (F) R  O  R  A
C11F  04 47 41 53 52 41 04 48 41 53 4C 41 04 49 52 4F
```



color review

"Personal Finance" ROM Pak program for the TRS-80 Color Computer
by The Image Producers, Inc.
distrubuted by Tandy Corporation – cat. no. 26-3101
$29.95

There is a special place in my heart for Radio Shack's "Personal Finance" program for the TRS-80 color computer. HOWEVER, it is not the same place that I reserve for things truly useful and logical.

I am relatively new to the ranks of real-live owners of computers and I owe thanks for that, in part, to the Personal Finance program. We are not long on budget around our house and any purchase the fiscal size of a computer must be justified to the family comptroller. Thanks to the well-known "local Radio Shack dealer," I was able to borrow a computer and aforementioned Personal Finance program for the purpose of convincing my wife that a computer at home is a very reasonable thing indeed. From what I have told you before you may assume that the ploy was successful. It was. The program not only demonstrated the usefulness of the machine, but after inputting some figures from our own personal finances it showed that purchase of the machine was feasible. Bless its little ROM.

Of course, with the purchase of the computer we picked-up the Personal Finance ROM pak and have used it since.

Now, you may ask just who I think I am, being a rank amateur and all, attempting to review a program for you, the sophisticated and experienced program-writing audience of *'68' Micro Journal*. Well, I am the guy the thing was 'designed' for. When we purchased this program neither my wife nor I had any real hands-on experience with computers and we both thought (as we were supposed to) that this program would make handling our budget easier. In its way it did. (Credit must be given where credit is due.) When we got all the numbers in, and all the goofs and confusion straightened out, it gave us a pretty clear (but not always pretty) all-in-one view of our financial situation. But, it took four months of experience with it to get all the numbers to come out right the first time through.

Perhaps we expected to much, but I think not.

A "Personal Finance" program, it seems to me, should be at least as easy to use as it is easy to reconcile your monthly checking statement. This one is fairly easy to operate—providing you get *every* entry right the first time (not much provision has been made for correcting mistakes), but it's not as easy as "doing the statement."

Any computer program should be "forgiving." It should allow some lee-way for the user. A program written for the inexperienced user should be, to my mind, very forgiving. The only bow to user oddities in the Personal Finance program is when selecting the month; i.e., where the programmer prefers to see "01" for January, it will accept simply "1". Elsewhere there is only one way to do it. Often enough to be irritating, it is the long way.

For instance, whenever you enter a number (as in dollars and cents) you must forget that the decimal point exists but never forget the two places to the right that are normally used in monetary functions—to enter $300 you *must* key 30000. This means an additional two keystrokes that *must* be remembered where quite often they shouldn't be needed. If you forget, and input it the way your shirt-pocket calculator lets you do it,

you'll end up with $3 entered and missed and then you'll have to drudge your way through the entire file to find where the heck the $297 error is.

Now, to find your mistake, you must restart the section of the program known as "Balance Checkbook." Next, do two steps for drill—tell it that you had no interest charges (a.k.a., services charges or check fees) and no interest earned. Now you may look at every uncancelled transaction in the program file. This is accomplished by *NOT* cancelling each item in turn until, hopefully, you find the bad entry.

If, when you find your error, it turns out that it is an uncancelled check, you're in luck. In this case a simple procedure allows you to change the value of the check and the program will update all affected balances.

However, if it is anything but an uncancelled check, you won't find it here or any place else in your program files. It is gone from the machine forever, except as figured into other budget balances. You are now limited to two time-consumming and usually complex choices—start your whole year's listing over from the beginning (forget it) or lie to your computer (a fearful and guilt-ridden experience, as you all know).

If you opt for the lie, then you still have to find the error. Doing your statement the old standard way is helpful during the investigative process, as is the assistance of a printing calculator. (There is no output to printer facility with this program.) Of course, while you're doing all this you start to wonder why.

When you finally have the mistake in hand, you must figure the difference between the figure in the files and the figure you figure should be the figure in the files. Then, to make up that difference in your files, you need to make up a fake transaction and tell the computer all about it. This involves starting the program section called "New Checks," entering the fake, exit "New Checks," start "Balance Checkbook" (no dear, still no charges or interest, thank you), and *NOT* cancel all transactions down to the fake, THEN CANCEL IT!

Other little things that are not quite right—a genuine bug exists in the input from/output to tape section. When you call for these functions your tape machine is turned on immediately and not when you are told it is, sometime later. You can live with this if you don't press "play"on your recorder when you are told to either. The program doesn't really start to output or input until you press enter. If your tape is blank when you want to output there is no real problem. But if you start to output in the middle of last month's data, then you get goo-gaa when you try to input it next month. Same is true if you're slow on the enter key during input. The tape has already run to the middle of the data when the machine starts to read. Fortunately, all data is automatically written three times to tape, so if at first you don't succeed, try, try, try again.

The fact that you can't output to printer with the stock program, as mentioned previously, is rather short-sighted on the part of the author and Tandy. This means that individual transaction items that are removed from the files for purposes of memory conservation are lost forever except in any hand-written records you may make while you are running the program. This does not make personal finance record keeping easier.

Other irritants in the program are that it keeps some things you don't want and won't keep others that you do.

While doing the initial set-up of budget files, all items are changeable to fit your needs. This is very good and as it should be. It is time consumming though, and since it is personal, you're doing it at home on your day off. By the time you get around to inputting the bank info (step 2 in the set-up procedure), you may be, as we were, a little fuzzy and anxious to get out to play. We should have gone on out—the bank info section is not so flexible. In the process of understanding just what it was the instructions were trying to get across, we made a few errors. Fortunately for us, the program allows you several bank accounts because it doesn't allow changes in important areas. It will let you to change the name and account number but nothing else. Nor can you delete an account. We got everything right from account no.3 on, but now we have accounts 1 and 2 at the XXX bank at least until next January.

As I said before, credit must be given where credit is due. The Personal Finance program will assist you in setting up a household budget and when the blanks are filled in properly each month (which, if done carefully, is nearly as easy as reconciling your checking account) you wind up with much more information than just doing the statement will give you. You will, after entering new checks and transactions for the month and cancelling the ones that came back with your statement, get a very clear picture of where you are doing well at keeping within your budget and just where it is you're slipping. If I may make an analogy, as in mountain climbing, the climb is a little rough but the view is worth it.

Michael Pepper

THOSE WONDERFUL MEMORY-MAPPED VIDEO BOARDS

PART II: GRAPHICS

THOMAS H. HUNT
30001 WAGNER
WARREN, MI 48093

To me,one of the most fascinatinn uses of the home computer is in the area of computer graphics. I have long yearned for the ability to create complex drawings, plots, schematics, and -- yes -- even animated games on my home computer. Dull text output could even be spruced up a bit by sprinkling in some special effects here and there.

This affinity for graphics is no accident. Mankind, in general, tends to think or create in terms of patterns and images. Conversly, we tend to comprehend more quickly and completely when information is presented to us in the form of images.

Graphics, therefore, should be the preferred human interface to a computer. This should not come as a new revelation to anyone. Hardware realizations go back to 1953 when the first graphics display terminal appeared at MIT as part of the WHIRLWIND Computer. In the almost three decades since this modest beginning, the computer graphics industry has exploded into an unbelievable level of sophistication. Commercial systems are available today with resolution that rivals the finest of color photographs. Even the ubiquitous video aracades have displays that I would be more than happy to own. And no manufacturer with any sense would attempt to market a ready-built computer today without some form of built-in graphics capability.

So now the burning question becomes -- How do I,as an SS-50 buss fnn, get my feet wet in this very rapidly changing and fairly expensive area? Ideally, I would like to get a board that was inexpensive, expendible if necessary, to act as a "training module". It could be black and white, for now, but should have sufficient resolution (256 x 192) to produce an interesting display. Individual pixel control is also a must -- TRS-80 style block graphics just does not appeal to my artistic side. And it should be easily constructed, perhaps by modifying an existing board.

But most important, this imaginary board should serve to educate me in the complexities and realities of the pixel graphics world. Hopefully it would bring my level of sophistication up to a point where I could be more discerning when spending big bucks for a first class graphics board. If it turns out that no such board is available, I would like my imaginary board to be sufficiently challenging to hold my interest until one does become available.

Now we get down to brass tacks --- converting this board into reality. First, I'll describe the hardware. While you are waiting for parts, you might as well type in the software and debug it. That's right -- I said debug. In today's changing world, it would be a waste of time to write software that was too hardware dependent! Just temporarily assign a spare 6K block of RAM as screen memory, and then view segments to see if the right things are happening.

HARDWARE

CHOOSING THE BOARD --

There are a multitude of schematics for graphics boards available but I, for one, am not too thrilled at the thought of wire-wrapping several hundred connections -- especially if there is an easier way. It turns out that the F&D Associates PMB-1 Video Board is quite easily modified for full pixel graphics. So the first thing to do is order one these boards from F&D -- honest, I don't get any kind of cut on the profits!

There are a couple of good reasons for chosing the F&D board. First and foremost, it uses the 6845 CRT Controller chip which has sufficient register range to generate a 256 x 256 graphics display (we will only implement 256 x 192). Even better, it is inexpensive. I got the board up and running for under $150. Best of all, when a better graphics board becomes available,the F&D board is just as easily converted back to a memory-mapped terminal board.

The modifications are quite easy and can be done in a couple of evenings. In further discussions, I will be referring to the F & D schematic -- first by their page number, then by their IC number (eg S6-IC40).

DOT CLOCK MODIFICATIONS

32 bytes are needed to clock out 256 pixels on one horizontal scan line. As 80 bytes are normally clocked out in alphanumeric applications, the dot clock must be slowed down to fill the screen in graphics applications. The easiest way to do this is to install a 6 to 7 MHz. crystal in place of the one called for by F&D (S7-CRY).

CHARACTER GENERATOR BYPASS

In alphanumeric applications, the data in screen memory is actually used as addresses for character data contained in a character generator ROM. Therefore, it is the output of the character generator that is really clocked out to the CRT screen. In a graphics application, what we really want is to clock out the direct contents of screen memory -- bit by bit. To accomplish this we must bypass the screen memory data lines around the character generator.

Referring to page S6 of the F&D schematic, do not install IC's 31, 38, and 41. The two zeners and their resistors are also not required. Next, add all three jumpers, J4, J5, and J6. In the holes vacated by the deleted ICs, nine jumpers need to be added. TABLE I lists these jumpers.

****

Actually, you can quit now if you are willing to accept 256 x 128 resolution (which is not too shabby for an hour's extra work). Just add the rest of the components -- sans PIA -- per the F&D instructions and proceed to the software and checkout phases. It might be a good idea to check the board out now, anyway. The software will work, but the bottom third of the screen will be a duplication of the top third due to memory address wraparound.

The rest of the hardware section deals with adding more screen memory to increase resolution. 256 x 192 pixels will require 6K of screen memory. 4K is already available on the F&D board, so we need to add an additional 2K of 2114's plus some appropriate decoding.

****

MEMORY ADDITION

To add 2K of memory, plus decoding, will require four 2114's, one 74157, and one 74138 (which replaces S2-IC20, a 74139). Refer to Figure 1 for the wiring details. This is the tricky portion of the modification and I hesitate to recommend any one particular method of construction.

For the memory chips,I mounted 6 wire wrap sockets on a perf board -- two of the sockets were long lead types and mounted to align with IC10 and IC11. Daisy chain the appropriate data, address, and power pins, then plug the assembly into the IC10,11 holes. VA12 is jumpered to the added 74157 (piggybacked on IC22). The chip selet lines are jumpered to the added 74138, which I simply cross-wired into the old IC20 location.

HARDWARE CHECKOUT

After appropriate address selection,plug the board into the buss and hook up your CRT monitor. If possible try to get a separate monitor for this phase. Upon power up, a raster should appear. As the 6845 has not been initialized,it will be a random, unsynced pattern.

The initialization routine should now be executed once. The display should now be square, stable and centered. The pattern will be whatever bit pattern happened to be in memory at power up. Filling screen memory with all $FF's will give a solid display and all $00'S will give a blank display.

Now is the time to play with the initialization values to obtain a centered, stable display. The initialization values given in the listing are valid for a 7.16 MHz crystal. F & D provides a sample calculation if you need to change any parameters. Basically, we are trying to initialize for 32 horizontal characters, one scan line per row, and 192 rows displayed. Interlaced mode must be used to allow sufficient range for the rows displayed register. Since we are calling for only one scan line per row, no actual interlacing will be apparent and the display will not flicker.

SOFTWARE

It does not take one too long, hand entering byte patterns into screen memory, to realize that this is definitely not the way to fly. Some more powerful and faster method to handle screen data will be required. Obviously, a software package of graphics drivers will be absolutely necessary if anything meaningful is to be accomplished.

A considerable amount of work has already been done (see references) on what a graphics package should be and algorithms to accomplish the tasks. Suffice to say that anyone even semi-serious about graphics should dig into this literature. I have tried to include in the references books and articles that are pertinent as well as readily available. So, rather than reinvent the wheel here,I will only state two important concepts about any graphics package.

First, the programmer's interface to the display area should be machine independent and familiar. There are several good methods, but the most popular is the X, Y Cartesian coordinate system. Thus the programmer should view the screen as a set of points on an X, Y grid.

Second, the software should be relatively independent of the actual graphics generating hardware. Today with different boards appearing like popcorn, it would be comforting to know that many hours of software development will not go completely down the tube. This is most commonly accomplished by designing the software in "layers". Only the most primitive layer, usually the pixel set/reset routine,actually "talks" to the particular hardware. Thus updating the graphics hardware will usually mean only changes to one software routine.

THE PIXEL ROUTINE

The graphics package in listing 1 is built around one "core" routine called PIXEL. PIXEL is the routine that actually performs the manipulation of the data on the screen in terms of X, Y coordinates. It is dependent upon the F & D hardware, or a similar type linear addressed, memory mapped board. As written, PIXEL is not relocatable, but the source may be reassembled to a different location. However, it must be ORG'ed at a 256 byte boundary.

To use PIXEL,the programmer views the screen as an X, Y grid with the origin (0,0) in the upper left hand corner of the screen. Many experts hotly argue that the origin should be in the lower left hand corner, giving the more familiar first quadrant view. One could also make a good case for having the origin in the center of the screen, especially if equation plotting is the main function desired.

Parameters are passed to PIXEL in three base page registers -- XBEG, YBEG,and MODE. Begin by loading the desired X,Y coordinate into XBEG and YBEG, respectively. Note that these registers are one byte, allowing the software to handle displays up to 256 x 256 resolution. Now that the point is defined, the MODE register tells PIXEL what to do with it. If MODE=0, the dot will be turned on and if MODE=1, the dot will be turned off.

If MODE is any value other than 1 or 0, the dot will be complemented. Complement mode is seldom implemented in hobbyist graphics software, but it is a very valuable function. Simply stated, Complement mode toggles the selected bit, i.e. if set, it gets reset and vice-versa. What this provides for is a simple, yet effective, way to move figures or lines through each other without losing any points.

After the three registers are loaded, a simple JSR PIXEL completes the operation. The MODE register need only be changed when a different mode is desired. While no registers are preserved, PIXEL does not change XBEG or YBEG. Interface to BASIC or other high level languages can be accomplished in the same manner by using PEEK, POKE and USER routines.

The mechanics of how the PIXEL routine operates may not be readily obvious. PIXEL has to perform three distinct operations. First, as the hardware is linear addressed, the X,Y coordinate must be converted into an absolute screen memory address. The general formula for accomplishing this requires an 8 x 8 multiply. However, as the horizontal resolution is a power of two, this reduces to shifts and rotates to form a 16 bit address.

Now that the absolute address of the byte is known the desired bit within that byte must be found. So, an appropriate mask is selected from a table -- the offset into the table is the three least significant bits of the X coordinate. Finally, MODE is checked and, with the address and mask, the appropriate bit is operated upon.

INITIALIZATION ROUTINE

It seems that almost everything in the computer world needs a one-time initialization. The graphics hardware and software is no exception. The initialization routine in Listing 1 simply sets up some software variables and loads the 6845 registers. Run it once after power up or reset.

THE LINE ROUTINE

At the heart of any graphics software is its line drawing routine. A graphics package that draws no lines, or even more exasperating, just horizontal and vertical lines, is next to worthless. It should almost go without saying that the ability to draw a line between any two pairs of X, Y points is probably the most important graphics primative around.

There are a number of algorithms available that, with varying degrees of success, manage to generate a line. If memory serves me correctly, the algorithm I have implemented was developed at IBM in the mid-1960's. The algorithm is not only quite clever but it also generates a very good line for a raster display. It is particularly well suited for microprocessors as it requires no divides or multiplies.

I will not attempt a detailed description of the algorithm in this article. However, interested readers can find an exceptionally good explaination in ref. 3, with an implementation in 8080 machine code. Another description, not quite as detailed, appears in ref.4, along with an implementation in BASIC.

Basically, the line routine calculates the series of X, Y points necessary to construct the line and feeds these points to PIXEL. As both input and output are X, Y coordinates, this routine is conveniently hardware independent. The special cases of single point, horizontal, vertical, and diagonal lines are coded separately -- mostly to improve speed. With the exception of the single point exit, these routines can be removed and the main body of LINE will work just fine.

To draw a line, first set up MODE as in PIXEL. Then load the four end-point coordinates of the desired line into XBEG, YBEG, XEND, and YEND. A JSR LINE will draw the line between and including the end points. Note that, upon RTS, XBEG and YBEG are updated to equal XEND and YEND. Therefore, to draw a series of connected lines, it is only necessary to reload XEND and YEND between calls to LINE.

APPLICATIONS

The software drivers presented in this article provide an excellent cornerstone for a potentially powerful graphics package. With a little practice, you will see that PIXEL and LINE, alone, are very powerful tools. Couple this with a BASIC that has a user routine and it is very easy to develop a set of simple BASIC statements that will duplicate about 80% of the graphic functions found on the TRS-80, PET, and APPLE (sans color). I suggest reference #5 for starters. This book provides a dozen or so BASIC programs for APPLE computers, most of which I was easily able to duplicate using TSC BASIC. So far I have also been able to duplicate about twenty graphics oriented programs in back issues of the "other" computer magazines. An entirely new world has been opened!

Although BASIC allows complex numerical manipulations to be quickly programmed,it is definitely slow. To get maximum speed, assembly language will have to be used. Listing 2 is an assembly language demonstration program. It is nothing exotic, but it does show off the lines available.

TIMING CONSIDERATIONS

In graphics, the name of the game is rapid execution. PIXEL is the prime target -- the faster hardware and software can set an X, Y pixel, the better. The PIXEL routine presented here will set any X,Y point in about 70 machine cycles, which is quite fast for an 8-bit machine using linear addressed hardware. X, Y addressing capability would save about 30 cycles per pixel -- well worth it. Just for comparison, expensive commercial graphics equipment (using 16 to 32 bit CPU's and enhanced hardware) routinely achieve pixel setting times of 1 to 5 microseconds!

The LINE routine uses about 100 machine cycles one time overhead and 70 machine cycles per pixel overhead. Other algorithms can be coded in less cycles, but they usually end up drawing more pixels. I really do not lay any claims to being the " world's best graphics programmer " -- or even second best, for that matter. So, I would welcome comments from readers that can find ways to measureably improve the execution times.

DRAWBACKS

Although the board exceeded my expectations, it does have several serious shortcomings. The first and most objectionable is the lack of an adequate solution to the memory contention problem (discussed in Part 1 of this article). The excessive screen flicker (due to blanking during CPU access) during rapid screen updates is entirely unacceptable. Any future graphics boards must provide some form of synchronous clock,transparent memory accessing scheme. Solutions that provide for updates only during horizontal or vertical retrace are simply not adequate for graphics.

The next major problem is inherent in the 6845 -- linear addressing. In any successful graphics system, the PIXEL routine must execute as quickly as possible. More than half of my PIXEL's execution time is taken up just to convert an X,Y point into a linear address!

As a result of the above two problems, I have had difficulty in achieving acceptable animation. Static displays are quite good, but my fond dream of arcade-type animation still eludes me. Also hampering the animation solution is the availability of only one screen image plane.

I have also learned that simultaneous graphics and alphanumerics capability is extremely desirable. It is difficult to communicate with the computer, or annotate displays,when the screen is tied up playing with pixels.

One must either get another CRT monitor, write a character generator/handler program, or work blind into the computer. None of these solutions are very attractive.

F & D Associates, of course, is not to be blamed directly for these problems. After all, their board is being used for something other than design intent.

EXTENSIONS AND IMPROVEMENTS

In the course of working with this hardware and software, many improvements have occurred to me. A major hardware improvement would be to make the board into a pseudo alpha-graphics terminal. Nothing in the modifications precludes using the EPROM slot. Thus, a standard character generator could be programmed into the EPROM and plugged in. Only two additional control lines are now needed (perhaps using the onboard PIA). One control changes the clock rate and the other selects either direct screen memory data or EPROM data. We now have a software controlled alphanumerics OR graphics terminal.

The software extensions are limited only by the imagination. As a stopgap solution, a character generator routine could be written to annotate the display. It could even be linked to a terminal driver for full I/O. Some form of shape-drawing routine would also be highly desirable. End points of an arbitrary figure could be defined in RAM and drawn on the screen by the shape routine (or perhaps a series of unit vectors ala the APPLE).

Once points, lines, and shapes are available, more complex routines could be devised to operate on these basic elements. A Scaling Routine (change size but not shape) and a Translation Routine (move the element, but do not change size or orientation) would be two of the easier ones to write. Translation should also include the special case of Shear (change either X or Y,but not both) for interesting effects. A more complex routine would be that of rotation about any arbitrary point or axis. A Fill Routine -- given a beginning and end point fill in any bounded shape -- would also be useful.

The mathematics behind any of these routines is fairly straightforward and readily available (Ref. 6), but actual implementation on an 8-bit machine may require a little cleverness. Remember, as long as the output of these routines is only X,Y points that feed PIXEL or LINE, your efforts will not become obsolete. It would also be wise to make provisions for a COLOR register and the ability to handle greater than 256 x 256 resolution (the 68000 is coming faster than you think).

CONCLUSION

In this part of the article, I have presented the hardware and the minimum software necessary to create a 256 x 192 pixel graphics display for under $150. This was not intended to be a graphics panacea as should be evident by the problems I described. But it has served its primary purpose in providing an inexpensive, useful and educational introduction to graphics. It has certainly provided me with many hours of enjoyment. I hope this series of articles will stimulate further interest in graphics and,also, spark manufacturers into designing equipment to compete with the ready-built computers and home arcade games.

REFERENCES:

1. Rogers,D,and Adams,J.A, MATHEMATICAL ELEMENTS FOR COMPUTER GRAPHICS,McGraw-Hill,N.Y.,1976
2. Hungerford,J.C., GRAPHIC MANIPULATIONS USING MATRICES, Byte , September 1978
3. Jones,V, A PROPOSED GRAPHICS SOFTWARE STANDARD, Byte, Technical Forum, November 1979
4. Higgins,M., FAST LINE DRAWING TECHNIQUE, Byte, August 1981
5. Waite,M., COMPUTER GRAPHICS PRIMER, Howard Sams & Co.,Indianapolis,Ind.,1980
6. Posdamer,J. L., THE MATHEMATICS OF COMPUTER GRAPHICS, Byte, September 1978
7. Koritos,B.J.,GRAPHIC SOFTWARE FOR MICROCOMPUTERS, Kern Publications, Duxbury, Ma.,1981
8. Brown, B.E.,and Levine S., THE FUTURE OF COMPUTER GRAPHICS, Byte, November 1980
9. Killan, L., THE BASICS OF COMPUTER ART, Kilobaud-Microcomputing, December 1979
10. Knutson, J., A VIDEO GRAPHICS PRIMER, Kilobaud-Microcomputing, November 1980

	FROM		TO	SIGNAL
1.	IC31-4	-	IC31-5	D6
2.	IC31-6	-	IC31-9	D4
3.	IC31-7	-	IC31-12	D2
4.	IC31-8	-	IC31-19	D5
5.	IC31-11	-	IC31-18	D3
6.	IC31-16	-	IC31-17	D1
7.	IC31-20	-	IC38-2	D7
8.	IC31-15	-	IC29-12	D0
9.	IC29-11	-	IC38-6	D8

NOTES: 1. REMOVE IC31 AND IC38
2. ADD JUMPER IC41-9 TO IC38-6 TO USE IC41 AS CHAR. GEN.

TABLE I. Jumper list for F&D conversion

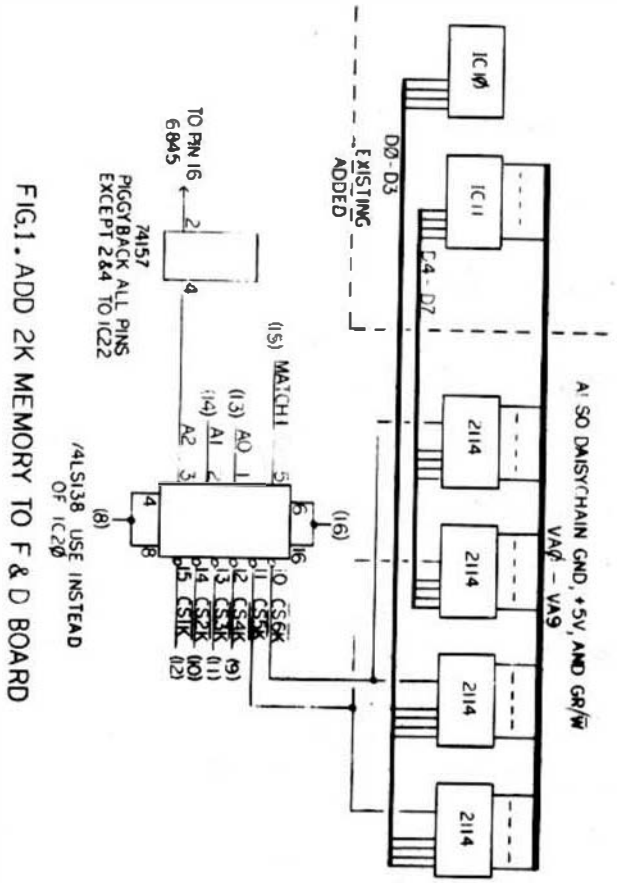


FIG.1. ADD 2K MEMORY TO F & D BOARD

```
   ---    PAGE  001   GRAPH

00001                        NAM     GRAPH
00002                        OPT     B,S,NOG
00003                  * WRITTEN BY TOM HUNT
00004                  * COPYRIGHT 1981
00005                  ********************************************************
00006                  *  THIS IS A SET OF PRIMITIVE GRAPHICS DRIVERS TO
00007                  *MANIPULATE A LINEAR-ADDRESSED VIDEO BOARD THAT
00008                  *HAS BEEN SET UP FOR INDIVIDUAL PIXEL CONTROL.  THE
00009                  *SOFTWARE IS CAPABLE OF HANDLING UP TO 256 X 256
00010                  *RESOLUTION.
00011                  *  THE PROGRAMMER INTERFACES TO THE DRIVERS BY
00012                  *VIEWING THE SCREEN AS AN X,Y CARTESIAN GRID WITH
00013                  *THE ORIGIN (0,0) IN THE UPPER LEFT HAND CORNER
00014                  *OF THE CRT.  THE FUNCTIONS PROVIDED ARE:
00015                  *    1. PIXEL -- SET ANY X,Y PIXEL
00016                  *    2. LINE -- DRAW A LINE BETWEEN TWO X,Y POINTS.
00017                  *       NOTE: MODE=1, PIXEL(S) WILL BE RESET.
00018                  *             MODE=0, PIXEL(S) WILL BE SET.
00019                  *             MODE)1, PIXEL(S) WILL BE COMPLEMENTED.
00020                  *    3. INIT -- INITIALIZE THE GRAPHICS BOARD.
00021                  *    4. SCREEN -- LOAD SCREEN MEMORY WITH ANY BYTE
00022                  *                 (USUALLY A CLEAR SCREEN).
00023                  ********************************************************
00024                  *  REVISION HISTORY:
00025                  *    VER 1.1   SPEED UP PIXEL
00026                  *    VER 1.2   SPEED UP LINE
00027                  *    VER 1.3   ADD COMPLEMENT MODE
00028                  *    VER 1.4   IMPROVE DRAW
00029                  *    VER 1.5   ADD FAST EXITS
00030                  ********************************************************
00031                  *
00032                  *  SYSTEM EQUATES
00033                  *
00034  9500            VIDADD EQU   $9500    6847 CONTROLLER
00035  0000            BASADD EQU   $00
00036  0000            VIDMEM EQU   $0000
00037  E7FF            SCREND EQU   $E7FF
00038                  *
00039                  ******************************
00040                  *    BASE PAGE VARIABLES     *
00041                  ******************************
00042                  *
00043 0000                   ORG     $0000
00044                  *
00045 0000 0001      XBEG   RMB     1         VECTOR COORDS
00046 0001 0001      YBEG   RMB     1
00047 0002 0001      XEND   RMB     1
00048 0003 0001      YEND   RMB     1
00049 0004 0002      MASK   RMB     2         ADDR OF PIXEL MASK
00050 0006 0001      XCOUNT RMB     1
00051 0007 0001      XMOVE  RMB     1
00052 0008 0001      YMOVE  RMB     1         INCREMENT VALUES(1,0, OR -1
00053 0009 0001      XCHANG RMB     1
00054 000A 0001      YCHANG RMB     1
00055 000B 0001      XINC   RMB     1
00056 000C 0001      YINC   RMB     1
00057 000D 0001      TEMPB1 RMB     1
00058 000E 0001      TEMPS2 RMB     1
00059 000F 0002      AMOV   RMB     2
00060 0011 0002      BMOV   RMB     2         XMOVE/2
00061 0013 0006      TEMP   RMB     6
00062 0019 0001      MODE   RMB     1
00063                  *
00064 7400                   ORG     $7400
00065                  *
00066                  ****************************
00067                  *   SYSTEM CONSTANTS       *
00068                  ****************************
00069                  *
00070 7400 80        MSKTBL FCB     $80,$40,$20,$10
00071 7404 01                FCB     8,4,2,1   BIT MASK TABLE
00072                  *
00073 7408 38        VIDTAB FCB     56        INIT CONSTANTS,7.2 MHZ CLOCK
00074 7409 20                FCB     32        AND 256 X 192 RASTER
00075 740A 2A                FCB     42,5,127,85,96
00076 740F 6F                FCB     111,3,1,32,0
00077 7414 00                FCB     0,0,0,0,0,0
00078                  *
00079                  ********************************
00080                  *     INITIALIZATION ROUTINE   *
00081                  ********************************
00082                  *
00083 741A 5F        INIT   CLR B             SET UP DISPLAY
00084 741B CE 7400           LDX     #MSKTBL   INIT TABLE POINTER
00085 741E DF 04             STX     MASK
00086 7420 CE 7408           LDX     #VIDTAB   LOAD 6845 REGISTERS
00087 7423 F7 9500 INIT1     STA B   VIDADD
00088 7426 A6 00             LDA A   0,X
00089 7428 B7 9501           STA A   VIDADD+1
00090 742B 08                INX
00091 742C 5C                INC B
00092 742D C1 12             CMP B   #18
00093 742F 26 F2             BNE     INIT1
00094 7431 4F        SCREEN CLR A
00095 7432 20 02             BRA     SCR1
00096                  * PUT VALUE IN TEMPB1, ENTER HERE TO FILL SCREEN
00097 7434 96 0D             LDA A   TEMPB1
00098 7436 CE 0000   SCR1    LDX     #VIDMEM
00099 7439 A7 00     SCR2    STA A   0,X
00100 743B 08                INX
00101 743C 8C E800           CPX     #SCREND+1
00102 743F 26 F8             BNE     SCR2
00103 7441 39                RTS
00104                  *
00105                  ********************************************************
00106                  *SET A PIXEL.  ENTRY POINT FOR  BASIC PGMS, POKE X &
00107                  *COORD INTO XBEG & Y COORD INTO YBEG.MODE=0.          *
00108                  *FIRST, CONVERT X,Y TO MEMORY ADDR.                   *
00109                  ********************************************************
00110                  *
00111 7442 96 01     PIXEL  LDA A   YBEG      FETCH X,Y
00112 7444 D6 00             LDA B   XBEG
00113 7446 44        PIXEL1 LSR A             MANIPULATE THE
00114 7447 56                ROR B           BITS
00115 7448 44                LSR A           UNTIL--
00116 7449 56                ROR B
00117 744A 44                LSR A           ACC A=MSB
00118 744B 56                ROR B           ACC B=LSB
00119 744C DB D0             ADD A   #BASADD  FORM ABSOLUTE ADDR.
00120 744E 97 0D             STA A   TEMPB1   AND SAVE IT
00121 7450 D7 0E             STA B   TEMPB2
00122                  *
00123                  *  NOW GET THE MASK
00124                  *
00125 7452 96 00             LDA A   XBEG
00126 7454 84 07             AND A   #7
00127 7456 97 05             STA A   MASK+1
00128 7458 DE 04             LDX     MASK     GET POINTER
00129 745A A6 00             LDA A   0,X      FETCH MASK
00130                  *
00131                  * CHANGE PIXEL IN REFRESH MEMORY
00132                  *
00133 745C DE 0D             LDX     TEMPB1   POINT TO SCREEN MEMORY
00134 745E D6 19             LDA B   MODE     SET OR RESET?
00135 7460 26 05             BNE     CLEAR
00136 7462 AA 00             ORA A   0,X
00137 7464 A7 00             STA A   0,X      SET MODE
00138 7466 39                RTS
00139 7467 C1 01     CLEAR   CMP B   #1
00140 7469 26 06             BNE     COMPL
00141 746B 43        CLEAR1  COM A            CLEAR MODE
00142 746C A4 00             AND A   0,X
00143 746E A7 00             STA A   0,X
00144 7470 39                RTS
00145 7471 16        COMPL   TAB              COMPLEMENT MODE
00146 7472 E4 00             AND B   0,X
00147 7474 26 F5             BNE     CLEAR1
00148 7476 AA 00             ORA A   0,X
00149 7478 A7 00             STA A   0,X
00150 747A 39                RTS
00151                  *
00152                  * VERTICAL LINE & SINGLE POINT FAST EXIT
00153                  *
00154 747B 96 03     VLINE   LDA A   YEND
00155 747D D6 01             LDA B   YBEG
00156 747F 10                SBA              YBEG-YEND?
00157 7480 26 02             BNE     VLINE1
00158 7482 20 BE             BRA     PIXEL    ONE POINT ONLY EXIT
00159 7484 22 04     VLINE1  BHI     VLINE2   DRAW DOWN
00160 7486 70 000A           NEG     YCHANG   DRAW UP
00161 7489 40                NEG A
00162 748A 97 0B     VLINE2  STA A   YMOVE
00163 748C BD B4             BSR     PIXEL    SET FIRST POINT
00164 748E 5F                CLR B            SET COUNTER
00165 748F 96 01     VLINE3  LDA A   YBEG     LOOP TO DRAW LINE
00166 7491 9B 0A             ADD A   YCHANG
00167 7493 97 01             STA A   YBEG
00168 7495 37                PSH B
00169 7496 BD AA             BSR     PIXEL
00170 7498 33                PUL B
00171 7499 5C                INC B
00172 749A D1 0B             CMP B   YMOVE
00173 749C 26 F1             BNE     VLINE3
00174 749E 39                RTS
00175                  *
00176                  * HORIZONTAL LINE FAST EXIT
00177                  *
00178 749F BD A1     HLINE   BSR     PIXEL
00179 74A1 5F                CLR B            RESET COUNTER
00180 74A2 96 00     HLINE1  LDA A   XBEG
00181 74A4 9B 0B             ADD A   XINC     UPDATE X
00182 74A6 97 00             STA A   XBEG
00183 74A8 37                PSH B
00184 74A9 BD 97             BSR     PIXEL    SET A POINT
00185 74AB 33                PUL B
00186 74AC 5C                INC B            BUMP COUNT
00187 74AD D1 07             CMP B   XMOVE
00188 74AF 26 F1             BNE     HLINE1
00189 74B1 39                RTS
00190                  *
00191                  * DIAGONAL LINE FAST EXIT
00192                  *
00193 74B2 BD 8E     DLINE   BSR     PIXEL
00194 74B4 5F                CLR B            SET COUNT
00195 74B5 96 00     DLINE1  LDA A   XBEG
00196 74B7 9B 0B             ADD A   XINC     UPDATE X
00197 74B9 97 00             STA A   XBEG
00198 74BB 96 01             LDA A   YBEG     UPDATE Y
00199 74BD 9B 0A             ADD A   YCHANG
00200 74BF 97 01             STA A   YBEG
00201 74C1 37                PSH B
00202 74C2 BD 7442           JSR     PIXEL
00203 74C5 33                PUL B
00204 74C6 5C                INC B
00205 74C7 D1 07             CMP B   XMOVE
00206 74C9 26 EA             BNE     DLINE1
00207 74CB 39                RTS
00208                  *
00209                  ********************************************************
00210                  *DRAW A LINE BETWEEN COORDS IN XBEG,YBEG AND XEND, &
00211                  *YEND.  SET MODES FOR DESIRED RESULT.                 *
00212                  ********************************************************
00213                  *
00214 74CC 4F        LINE    CLR A            INITIALIZE SOME VARIABLES
00215 74CD 97 0C             STA A   YINC
00216 74CF 97 09             STA A   XCHANG
00217 74D1 97 11             STA A   BMOV
00218 74D3 4C                INC A
00219 74D4 97 0B             STA A   XINC
00220 74D6 97 0A             STA A   YCHANG
00221 74D8 96 02     XLINE   LDA A   XEND
00222 74DA D6 00             LDA B   XBEG
00223 74DC 10                SBA              COMPUTE MAG OF X
00224 74DD 27 9C             BEQ     VLINE
00225 74DF 22 04             BHI     YLINE    AND MAG OF Y.
00226 74E1 70 0009           NEG     XINC     SET UP DX AND DY
00227 74E4 40                NEG A            ALONG THE WAY
00228 74E5 97 07     YLINE   STA A   XMOVE
00229 74E7 44                LSR A
00230 74E8 97 12             STA A   BMOV+1
00231 74EA 96 03             LDA A   YEND
00232 74EC D6 01             LDA B   YBEG
00233 74EE 10                SBA
00234 74EF 27 AE             BEQ     HLINE
00235 74F1 22 04             BHI     MAGCMP
00236 74F3 70 000A           NEG     YCHANG
00237 74F6 40                NEG A
00238 74F7 97 08     MAGCMP  STA A   YMOVE
```

```
00239 74F9 D6 07          LDA B  XMOVE
00240 74FB 11             CBA                    DRAWING WHICH WAY?
00241 74FC 27 B4          BEQ    DLINE
00242 74FE 23 15          BLS    LINE2
00243 7500 97 07          STA A  XMOVE           SWAP VECTORS
00244 7502 44             LSR A
00245 7503 97 12          STA A  BMOV+1
00246 7505 D7 08          STA B  YMOVE
00247 7507 96 0B          LDA A  XINC
00248 7509 97 09          STA A  XCHANG
00249 750B 7F 000D        CLR    XINC
00250 750E 96 0A          LDA A  YCHANG
00251 7510 97 0C          STA A  YINC
00252 7512 7F 000A        CLR    YCHANG
00253 7515 BD 7442 LINE2  JSR    PIXEL           SET 1ST POINT
00254                     *
00255                     * ALL IS READY, GO DRAW THE LINE
00256                     *
00257 7518 5F             CLR B
00258 7519 96 08    DRAW  LDA A  XBEG            ADD IN THE INITIAL
00259 751B 9B 19          ADD A  #INC            INCREMENTS
00260 751D 97 00          STA A  XBEG
00261 751F 96 01          LDA A  YBEG
00262 7521 9B 0C          ADD A  YINC
00263 7523 97 01          STA A  YBEG
00264 7525 96 12          LDA A  BMOV+1
00265 7527 9B 08          ADD A  YMOVE           BMOV=BMOV+YMOVE
00266 7529 97 12          STA A  BMOV+1
00267 752B 25 04          BCS    BOTH
0026  75 D 91 07          CMP A  XMOVE
00269 752F 23 10          BLS    DRAW1
00270 7531 98 07    BOTH  SUB A  XMOVE           INCREMENT BOTH VECTORS
00271 7533 97 12          STA A  BMOV+1
00272 7535 9A 00          LDA A  XBEG
00273 7537 9B 09          ADD A  XCHANG
00274 7539 97 00          STA A  XBEG
00275 753B 96 01          LDA A  YBEG
00276 753D 9B 0A          ADD A  YCHANG
002   753F 97 01          STA A  YBEG
00278 7541 37       DRAW1 PSH B                  SAVE COUNTER
00279 7542 BD 7442        JSR    PIXEL           SET A POINT
00280 7545 33             PUL B
00281 7546 5C             INC B
00282 7547 D1 07          CMP B  XMOVE
00283 7549 26 CE          BNE    DRAW
00284 754B 39             RTS
00285                     END

SYMBOL TABLE :

VIDADD=9500  BASADD=0000  VIDMEM=0000  SCREND=1FFF  XBEG  =0008
YBEG  =0001  XEND  =0002  YEND  =0003  MASK  =0000  SECPNT=0006
XMOVE =0007  YMOVE =0008  XCHANG=0009  YCHANG=000A  XINC  =000B
YINC  =000C  TEMPS1=000D  TEMPS2=000E  AMOV  =000F  BMOV  =0011
TEMP  =0013  MODE  =0019  MAKTBL=7400  VIDIAD=740B  INIT  =741A
INIT1 =742D  SCREEN=7431  CBI   =744A  SCN2  =7459  PIXEL =7442
PIXEL1=7446  CLEAR =74A7  CLEAR1=74AB  COMPL =7471  VLINE =749B
VLINE1=7484  VLINE2=74DA  VLINE3=74BF  HLINE =749F  HLINE1=74A2
DLINE =74B2  DLINE1=74B5  LINE  =74CD  XLINE =74D0  YLIMIT=74E5
MAGEMP=74F7  LINE2 =7515  DRAW  =7518  BOTH  =7531  DRAW1 =7541
TOTAL ERRORS 00000
```

```
00001                     APP    DEMO    GRAPHICS [illegible]
00002                     OPT    B,S,MOD
00003                     * WRITTEN BY TOM SAINT
00004                     * ALL RIGHTS RESERVED (C) I CAN SING
00005                     *  ANYONE FOOLISH ENOUGH TO PAY FOR IT
00006                     ********************************************
00007                     *  THIS PROGRAM DEMONSTRATES THE CAPABILITIES *
00008                     * OF THE SOFTWARE PACKAGE  "GRAPH"  WHEN USED *
00009                     * WITH A LINE88 ADDRESSED GRAPHICS BOARD.     *
00010                     * BEGIN EXECUTION AT START.  SIT BACK AND WATCH*
00011                     ********************************************
00012                     * REVISION HISTORY:
00013                     *    VER 1.0  8/12/81
00014                     ********************************************
00015                     *
00016                     *   SYSTEM LINKS
00017                     *
00018        0000   XBEG  EQU    $0000           BOTTOM CORNER
00019        0001   YBEG  EQU    $0001
00020        0002   XEND  EQU    $0002
00021        0003   YEND  EQU    $0003
00022        0019   MODE  EQU    $0019           MODE REGISTER
00023        741A   INIT  EQU    $741A           GRAPHICS INITIALIZATION
00024        7436   FILSCR EQU   $7436           FILL SCREEN
00025        7442   PIXEL EQU    $7442
00026        74CD   LINE  EQU    $74CD           DRAW LINE
00027        00BF   YMAX  EQU    191             MAX Y DIMENSION
00028        007F   XCEN  EQU    127             CENTER  SCREEN
00029        005F   YCEN  EQU    95
00030                     *
00031 0100                ORG    $0100
00032                     *
00033 0100 BD 741A  START JSR    INIT            SET UP BOARD
00034 0103 4F       START1 CLR A
00035 0104 BD 7436        JSR    FILSCR          CLEAR SCREEN
00036 0107 BD 0236        JSR    DELAY
00037                     *
00038                     ************************
00039                     * DEMONSTRATE LINES    *
00040                     ************************
00041                     *
00042 010A 7F 0019  DLINE CLR    MODE            SELECT "SET" MODE
00043 010D BD 5E          BSR    FAN             DRAW FIGURE
00044 010F BD 02 6        JSR    DELAY           DISPLAY PATTERN A WHILE
00045 0112 7C 0019        INC    MODE            SELECT "RESET"
00046 0115 80 5           BSR    FAN             ERASE IT
00047 0117 BD 0236        JSR    DELAY
00048                     *
00049                     ************************
00050                     * DEMONSTRATE SET MODE *
00051                     ************************
00052                     *
00053 011A 7F 0019  SET   CLR    MODE
00054 011D BD 0211        JSR    SQUARE          DRAW SQUARE
00055 0120 BD 0236        JSR    DELAY
00056 0123 BD 01D7        JSR    FILLFT          RUN CURTAIN
00057 0126 BD 02 6        JSR    DELAY
```

```
00058 0129 7C 0019        INC    MODE            BACK AND
00059 012C BD 01F2        JSR    FILRGT          FORTH
00060 012F BD 0236        JSR    DELAY
00061                     *
00062                     ************************
00063                     * DEMONSTRATE RESET MODE *
00064                     ************************
00065                     *
00066 0132 86 FF    RESET LDA A  #$FF
00067 0134 BD 7436        JSR    FILSCR          FILL  SCREEN
00068 0137 BD 0236        JSR    DELAY
00069 013A BD 0211        JSR    SQUARE
00070 013D BD 0236        JSR    DELAY
00071 0140 BD 01D7        JSR    FILLFT
00072 0143 BD 0236        JSR    DELAY
00073 0146 7F 0019        CLR    MODE
00074 0149 BD 01F2        JSR    FILRGT
00075 014C BD 0236        JSR    DELAY
00076                     *
00077                     ************************************
00078                     * DEMONSTRATE COMPLEMENT MODE        *
00079                     ************************************
00080                     *
00081 014F 4F       COMPL CLR A
00082 0150 BD 7436        JSR    FILSCR          CLEAR SCREEN
00083 0153 BD 0236        JSR    DELAY
00084 0156 BD 0211        JSR    SQUARE
00085 0159 BD 0236        JSR    DELAY
00086 015C 7A 0019        DEC    MODE            SELECT COMPLEMENT
00087 015F BD 01D7        JSR    FILLFT
00088 0162 BD 0236        JSR    DELAY
00089 0165 BD 01F2        JSR    FILRGT
00090 0168 BD 0236        JSR    DELAY
00091 016B 20 96          BRA    START1          LOOP FOREVER
00092                     *
00093                     ************************
00094                     * DRAWING SUBROUTINES  *
00095                     ************************
00096                     *
00097 016D B6 7F    FAN   LDA A  #XCEN            GET TO CENTER SCREEN
00098 016F 97 02          STA A  XEND
00099 0171 B6 5F          LDA A  #YCEN
00100 0173 97 03          STA A  YEND
00101 0175 86 C8          LDA A  #$C8
00102 0177 B7 0245        STA A  YLIMIT
00103 017A 4F             CLR A                  SET UP FIRST LINE
00104 017B B7 0243        STA A  YSAVE
00105 017E 97 01          STA A  YBEG
00106 0180 86 04          LDA A  #4              SPACE BETWEEN LINES
00107 0182 B7 0244        STA A  INCR
00108 0185 B7 0242        STA A  XSAVE
00109 0188 97 00          STA A  XBEG
00110 018A BD 10          BSR    FAN1            DRAW IT
00111 018C B7 0243        STA A  YSAVE
00112 018F 97 01          STA A  YBEG
00113 0191 B6 0242        LDA A  XSAVE
00114 0194 97 00          STA A  #BEG
00115 0196 7F 0245        CLR    YLIMIT
00116 0199 70 0244        NEG    INCR            ### "COM" FOR UNUSUAL EFFECT
00117 019C BD 74CD  FAN1  JSR    LINE            DRAW A LINE
00118 019F B6 0243        LDA A  YSAVE           UPDATE Y
00119 01A2 97 01          STA A  YBEG
00120 01A4 B6 0242        LDA A  XSAVE           UPDATE X
00121 01A7 BB 0244        ADD A  INCR
00122 01AA 97 00          STA A  XBEG
00123 01AC B7 0242        STA A  XSAVE
00124 01AF 26 ER          BNE    FAN1
00125 01B1 B0 0244        SUB A  INCR            FIX UP FOR SIDES
00126 01B4 97 00          STA A  XBEG
00127 01B6 B7 0242        STA A  XSAVE
00128 01B9 B6 0243        LDA A  YSAVE
00129 01BC 97 01          STA A  YBEG
00130 01BE BD 74CD  FAN2  JSR    LINE            DRAW SIDES
00131 01C1 B6 0242        LDA A  XSAVE
00132 01C4 97 00          STA A  XBEG
00133 01C6 B6 0243        LDA A  YSAVE
00134 01C9 BB 0244        ADD A  INCR
00135 01CC B7 0243        STA A  YSAVE
00136 01CF 97 01          STA A  YBEG
00137 01D1 B1 0245        CMP A  YLIMIT
00138 01D4 26 EH          BNE    FAN2
00139 01D6 39             RTS
00140                     *
00141                     * FILL SCREEN, LEFT TO RIGHT
00142                     *
00143 01D7 4F       FILLFT CLR A                  SET UP LINE
00144 01D8 97 00          STA A  XBEG
00145 01DA 97 01          STA A  YBEG
00146 01DC 97 02          STA A  XEND
00147 01DE 86 BF          LDA A  #YMAX
00148 01E0 97 03          STA A  YEND
00149 01E2 BD 74CD  FILOOP JSR   LINE            FILL SCREEN LOOP
00150 01E5 7F 0001        CLR    YBEG            UPDATE LINE COORDS
00151 01E8 96 00          LDA A  XBEG
00152 01EA 4C             INC A
00153 01EB 97 00          STA A  XBEG
00154 01ED 97 02          STA A  XEND
00155 01EF 26 F1          BNE    FILOOP
00156 01F1 39             RTS
00157                     *
00158                     * FILL SCREEN, RIGHT TO LEFT
00159                     *
00160 01F2 7F 0001  FILRGT CLR   YBEG            SET UP LINE
00161 01F5 86 FF          LDA A  #$FF
00162 01F7 97 00          STA A  XBEG
00163 01F9 97 02          STA A  XEND
00164 01FB 86 BF          LDA A  #YMAX
00165 01FD 97 03          STA A  YEND
00166 01FF BD 74CD  FILOP1 JSR   LINE
00167 0202 7F 0001        CLR    YBEG            UPDATE LINE COORDS
00168 0205 96 00          LDA A  XBEG
00169 0207 4A             DEC A
00170 0208 97 00          STA A  XBEG
00171 020A 97 02          STA A  XEND
00172 020C 81 FF          CMP A  #$FF     DONE?
00173 020E 26 EF          BNE    FILOP1
00174 0210 39             RTS
00175                     *
00176                     * DRAW A BIG SQUARE
```

```
00177                   *
00178 0211 B6 3F    SQUARE LDA A  #63       SET UP END POINTS
00179 0213 97 03           STA A  YEND
00180 0215 97 01           STA A  YBEG
00181 0217 B6 55           LDA A  #85
00182 0219 97 00           STA A  XBEG
00183 021B B7 0242         STA A  X AVE
00184 021E B6 AA           LDA A  #170
00185 0220 97 02           STA A  XEND
00186 0222 BD 74CD SQLOOP  JSR    LINE     DRAW LOOP
00187 0225 B6 0242         LDA A  XSAVE    UPDATE
00188 0228 97 00           STA A  XBEG
00189 022A 96 03           LDA A  YEND
00190 022C 4C              INC A
00191 022D 97 03           STA A  YEND
00192 022F 97 01           STA A  YBEG
00193 0231 81 7F           CMP A  #127     DONE?
00194 0233 26 ED           BNE    SQLOOP
00195 0235 39              RTS
00196                   *
00197 0236 CE 03FF  DELAY  LDX    #$03FF   DELAY A WHILE
00198 0239 86 FF    DEL1   LDA A  #$FF
00199 023B 4A       DEL2   DEC A
00200 023C 26 FD           BNE    DEL2
00201 023E 09              DEX
00202 023F 26 F8           BNE    DEL1
00203 0241 39              RTS
00204                   *
00205                   ********************************
00206                   *  TEMP STORAGE OF VARIABLES   *
00207                   ********************************
00208                   *
00209 0242 0001  XSAVE  RMB    1
00210 0243 0001  YSAVE  RMB    1
00211 0244 0001  INCR   RMB    1
00212 0245 0001  YLIMIT RMB    1
00213                   END

SYMBOL TABLE :

XBEG  =0000  YBEG  =0001  XEND  =0002  YEND  =0003  MODE  =0019
INIT  =741A  FILSCR=7436  PIXEL =7442  LINE  =74CD  YMAX  =00BF
XCEN  =007F  YCEN  =005F  START =0100  STARTI=0103  DLINE =010A
SET   =011A  RESET =0132  COMPL =014F  FAN   =016D  FAN1  =019C
FAN2  =01BE  FILLFT=01D7  FILOOP=01E2  FILRGT=01F2  FILOP1=01FF
SQUARE=0211  SQLOOP=0222  DELAY =0236  DEL1  =0239  DEL2  =023B
XSAVE =0242  YSAVE =0243  INCR  =0244  YLIMIT=0245
TOTAL ERRORS 00000
```

5240 S. W. Dosch Rd.,
Portland, Oregon 97201

Dear Don:

Here's a disk utility for FLEX 1.0 that complements the DISKSAVE routine by John Champlain in the August 1981 issue of '68' Micro Journal.

DISKFIX does this:

(a) Displays or prints a "complete" catalog listing (all catalog info);

(b) Checks a disk for a properly linked boot loader;

(c) Checks each disk file for proper record count and sector linkage and for agreement with the directory;

(d) Checks for file collisions;

(e) Checks the free-sector chain for proper linkage, etc.;

(f) Can (optionally) reconstruct the free-sectors chain from a map of the disk, constructed in memory during the previous operations.

The program is targeted at the 6800/DMAF-1 or 6800/DMAF-2 SWTPC systems, but is probably adaptable to most other FLEX systems.

The program has a few "peculiars" that need explanation. There are about 90 bytes of code dedicated to the handling of disks containing two kinds of files: (a) linked FLEX records and (b) block-type records not containing linkage data in the record, and having a different type of directory entry. These block-type records are coded for read/write/delete protection, and so can be recognized by DISKFIX (FLEX cannot ordinarily set the read-protect bit).

The program runs a bit longer than necessary because of some unused code in the MINIOPS section. MINIOPS is the disk-driver part of the subsidiary operating system MINIDOS which creates and handles those block-type program and data files.

Originally, the program was intended to reside at $A100-$A6FF, FLEX' utility command space. As finally configured, DISKFIX wouldn't fit, and so was moved to $5000. Since deleting all the code dealing with non-FLEX files and all the unused MINIOPS functions still does not trim away the necessary 3175 bytes, I just left the program the way it was known to work correctly.

If you can use this in '68' Micro Journal, good-nh. If not, return postage is enclosed.

DISKFIX PROGRAM

Geoffrey A. Glass
5240 S. W. Dosch Rd.,
Portland, Oregon 97201

DESCRIPTION

DISKFIX is a utility command for FLEX* 1.0 for the SWTPC 6800/DMAF-1 computer system. DISKFIX verifies the integrity of data and program files on a disk by tracing the directory entries and the chain of free sectors, and mapping the used sectors. It may be used to reestablish a valid chain of free sectors on a crashed disk.

DISKFIX detects linkage and record-count errors in the files on the disk, and reports any collision between files (two or more files linking to a common sector).

DISKFIX provides for skipping non-FLEX files on the disk if they are flagged with write/delete/read protection attributes codes. The program will not attempt to assign free sectors beyond the start of the first file of this type.

DISKFIX cannot check or restore a disk whose directory has been totally destroyed. It can verify the integrity of those files still having valid directory entries, and -- if desired -- open up the rest of the disk for re-use.

The printout of directory data and validation can be directed to the terminal or to the printer for a permanent record.

The program makes use of a block-oriented sub-operating system called MINIOPS, which uses FLEX primitives to address directly specific tracks and sectors. DISKFIX also relies heavily on FLEX registers and routines.

The program occupies addresses $5000-$5774, and uses the memory space $5800-$6FFF for a disk map and data storage. Page 0 addresses $0000-$0023 are also used by MINIOPS and the DISKFIX program. Except for the Page 0 references, the program and data areas can be relatively easily relocated to accommodate smaller systems not having memory in the $5000-$7FFF area.

OPERATING INSTRUCTIONS

To call DISKFIX, insert a disk carrying the DISKFIX command in one of the drives. For printer output, valid PRINT.SYS and P.CMD files must be available on the disk, and any additional printer driver routines required must be resident in memory.

There are four ways to invoke DISKFIX, in response to FLEX' +++ prompt:

Keyboard Input	Checks disk in	With output to
DISKFIX 0	Drive 0	Terminal
DISKFIX 1	Drive 1	Terminal
P DISKFIX 0	Drive 0	Printer
P DISKFIX 1	Drive 1	Printer

If the drive number is omitted, DISKFIX will prompt for it.

DISKFIX first looks at the "System Record" data on Track 0 Sector 3 of the indicated disk, and outputs the disk name and number, the start, end and sector-count of the free-sector chain, the disk creation date, and the disk address of the highest physical sector on the disk (4C1E for a double-sided disk).

> NOTE: If the system record does not show either 4C1E or 4C0F as the last entry, DISKFIX will be unable to generate a valid disk map, and consolidation should not be attempted until the System Record is corrected.

DISKFIX then will proceed to check Track 0 Sector 1 for a Boot Loader routine (the program that is loaded by the ROM monitor in response to the "O" command, and which loads DOS if DOS has been linked to the loader).

If there is no loader, or if it has not been linked to a disk address, the program will so note. If the Boot Loader register shows a non-zero disk address, it will be printed.

The program will then proceed to output each directory entry in a more complete form than the CAT command provides, numbering the entries consecutively for future identification.

For each directory entry, it will print the number, filename, extension, attributes, 00, start, end, length, sector-map

flag and date, plus either "O.K." or a description of some fault found as it attempted to trace the file.

The program reads the start of each sector of the file, noting the linkage to the next sector and the record number, and entering the file number in a disk "map" constructed in memory. If two programs link to the same sector, the collision is noted (File number and track and sector). Other errors detected are:

(1) Linkage 00 before the directory's sector-count is exhausted: "Link error at ttss".

(2) Record number out of sequence: "Record # error at ttss".

(3) Sector count exhausted, but linkage not 00: "Link error at ttss".

(4) Sector count 0, record # correct, linkage 00, but disk address disagrees with "end" address in directory: "Directory error".

The program will run 15 lines (terminal) or 48 lines (printer) and halt.

Press ESC to continue data output (or press RETURN to abort the program and return to DOS).

After all catalog entries have been checked, the program will report:

"Free sector chain: ", and check out the free sectors (this may take several minutes). It will then print either "O.K." or one of the error messages above.

Then it will report "Sector Map: " and will step through the constructed disk map, counting the free sectors. If this agrees with the system record count, the number will be printed.

If a write/delete/read protected file was encountered, the program will report the presence of any FLEX file linkages above the lowest such file, as well as the free sectors available below that disk address.

If the free-sector count disagrees with the system record, this is reported as a "Sector count error", followed by the count obtained by search of the map (the system record count was shown at the start of the output).

The program will then ask:

"Consolidate free sectors?". If the keyboard response is No, the program will exit to FLEX.

If the operator response was Yes, the program will construct a new free-sectors chain from its sector-map, up to $4C1E or to the limit set by the write/delete/read protected files. The sectors will all be numbered and re-linked, and the System Record will be given new start, end and length figures.

The program will then display "Oone." and exit to FLEX.

```
GOISK       #4  2217 2209  1378  07 25 00  4C1E

Boot Loader linked to $100
```

```
 1  PRINT    .SYS  00 00 0101 0101 0001 00 00 07 25 00 O.K.
 2  ERRORS   .SYS  00 00 0102 010A 0009 02 00 07 25 00 O.K.
 3  DOS      .SYS  00 00 0108 0205 0019 00 00 07 25 00 O.K.
 4  CAT      .CMD  00 00 0206 0208 0003 00 00 07 25 00 O.K.
 5  ASN      .CMD  00 00 0209 0209 0001 00 00 07 25 00 O.K.
 6  SAVE     .CMD  00 00 020A 020B 0002 00 00 07 25 00 O.K.
 7  EXEC     .CMD  00 00 020C 020C 0001 00 00 07 25 00 O.K.
 8  PROT     .CMD  00 00 020D 020D 0001 00 00 07 25 00 O.K.
 9  PRINT    .CMD  00 00 020E 020F 0002 00 00 07 25 00 O.K.
10  P        .CMD  00 00 0210 0210 0001 00 00 07 25 00 O.K.

11  DELETE   .CMD  00 00 0211 0212 0002 00 00 07 25 00 O.K.
12  COPY     .CMD  00 00 0213 0217 0005 00 00 07 25 00 O.K.
13  RENAME   .CMD  00 00 0218 0218 0001 00 00 07 25 00 O.K.
14  BUILD    .CMD  00 00 0219 0219 0001 00 00 07 25 00 O.K.
15  LIST     .CMD  00 00 021A 021C 0003 00 00 07 25 00 O.K.
16  TTYSET   .CMD  00 00 021D 021E 0002 00 00 07 25 00 O.K.
17  APPEND   .CMD  00 00 0301 0303 0003 00 00 07 25 00 O.K.
18  LINK     .CMD  00 00 0304 0304 0001 00 00 07 25 00 O.K.
19  DATE     .CMD  00 00 0305 0306 0002 00 00 07 25 00 O.K.
20  VERSION  .CMD  00 00 0307 0307 0001 00 00 07 25 00 O.K.

21  VERIFY   .CMD  00 00 0308 0308 0001 00 00 07 25 00 O.K.
22  I        .CMD  00 00 0309 0309 0001 00 00 07 25 00 O.K.
23  O        .CMD  00 00 030A 030  0002 00 00 07 25 00 O.K.
24  BACKUP   .CMD  00 00 030C 030E 0003 00 00 07 25 00 O.K.
25  NEWDISK  .CMD  00 00 030F 0314 0006 00 00 07 25 00 O.K.
26  JUMP     .CMD  00 00 0315 0315 0001 00 00 07 25 00 O.K.
27  QCHECK   .CMD  00 00 0316 0318 0003 00 00 07 25 00 O.K.
28  XOUT     .CMD  00 00 0319 031A 0002 00 00 07 25 00 O.K.
29  BASIC    .CMD  00 00 031B 0504 0026 00 00 07 25 00 O.K.
30  MEMTEST1.CMD  00 00 0505 0506 0002 00 00 07 25 00 O.K.
31  MPROT    .CMD  00 00 0507 0507 0001 00 00 07 25 00 O.K.
32  SYMSAV   .CMD  00 00 0508 0509 0002 00 00 07 25 00 O.K.
33  MINIDOS  .CMD  00 00 050A 0510 0007 00 00 07 25 00 O.K.
34  NEWDISK2.CMD  00 00 0511 0516 0006 00 00 07 25 00 O.K.
35  FORMAT3  .CMD  00 00 0517 051C 0006 00 00 07 25 00 O.K.
36  INSTFINO.BIN  00 00 051D 0602 0004 00 00 09 03 00 O.K.
37  DOITCK   .BIN  00 00 0603 0603 0001 00 00 09 04 00 O.K.
38  CORES001.TXT  00 00 0604 071A 0035 00 00 10 17 00 O.K.
39  CORES002.TXT  00 00 0718 0811 0015 00 00 10 17 00 O.K.
40  CORES003.TXT  00 00 0812 0907 0014 00 00 10 17 00 O.K.

41  CORES004.TXT  00 00 0908 091E 0017 00 00 10 17 00 O.K.
42  CORES005.TXT  00 00 0A01 0B05 0023 00 00 10 17 00 O.K.
43  CORES001.SYM  00 00 0BF6 0BF9 0004 00 00 10 17 00 O.K.
44  CORES002.SYM  00 00 0BFA 0B0B 0002 00 00 10 17 00 O.K.
45  CORES003.SYM  00 00 0B0C 0B0D 0002 00 00 10 17 00 O.K.
46  CORES004.SYM  00 00 0B0E 0B0F 0002 00 00 10 17 00 O.K.
47  CORES005.SYM  00 00 0B10 0B12 0003 00 00 10 17 00 O.K.
48  CORES001.BIN  00 00 0B13 0B16 0004 00 00 10 17 00 O.K.
49  CORES002.BIN  00 00 0B17 0B18 0002 00 00 10 17 00 O.K.
50  CORES003.BIN  00 00 0B19 0B1A 0002 00 00 10 17 00 O.K.

51  CORES005.BIN  00 00 0B1B 0B1D 0003 00 00 10 17 00 O.K.
52  CORES004.BIN  00 00 0B1E 0C01 0002 00 00 10 17 00 O.K.
53  CORES00 .BIN  00 00 0C02 0D02 001F 00 00 10 17 00 O.K.
54  CORES006.TXT  00 00 0D03 0D08 0006 00 00 10 17 00 O.K.
55  CORES006.BIN  00 00 0D09 0D09 0001 00 00 10 17 00 O.K.
56  CORES006.SYM  00 00 0D0A 0D0A 0001 00 00 10 17 00 O.K.
57  MAILER3 .BIN  00 00 0D0B 0D14 000A 00 00 10 23 00 O.K.
58  MAILER1 .BIN  00 00 0D15 0E04 000E 00 00 10 23 00 O.K.
59  MAILER2 .BIN  00 00 0E05 0E1C 0018 00 00 10 23 00 O.K.
60  MAILER4 .BIN  00 00 0E1D 0F06 0008 00 00 10 23 00 O.K.

61  MODXREF .BIN  00 00 0F07 1008 0020 00 00 11 10 00 O.K.
62  MINIOPS .TXT  00 00 1009 110C 0022 00 00 06 15 01 O.K.
63  MINIOPS .BIN  00 00 110D 110F 0003 00 00 06 15 01 O.K.
64  MINIOPS .SYM  00 00 1110 1112 0003 00 00 06 15 01 O.K.
65  MINIDOS1.TXT  00 00 1113 1217 0023 00 00 06 15 01 O.K.
66  MINIDOS1.BIN  00 00 1218 121A 0003 00 00 06 15 01 O.K.
67  MINIDOS1.SYM  00 00 1218 121E 0004 00 00 06 15 01 O.K.
68  MINIDOS2.TXT  00 00 1301 1401 001F 00 00 06 15 01 O.K.
69  MINIDOS2.BIN  00 00 1402 1404 0003 00 00 06 15 01 O.K.
70  MINIDOS2.SYM  00 00 1405 1407 0003 00 00 06 15 01 O.K.

71  MDOSMAN1.TXT  00 00 1408 1606 0038 00 00 06 15 01 O.K.
72  MDOSMAN3.TXT  00 00 1607 1617 0011 00 00 06 15 01 O.K.
73  MDOSMAN2.TXT  00 00 1618 1713 001A 00 00 06 15 01 O.K.
74  DISKFIX1.TXT  00 00 1F12 2018 0025 00 00 07 07 01 O.K.
75  DISKFIX1.BIN  00 00 1F0F 1F11 0003 00 00 07 07 01 O.K.
76  DISKFIX1.SYM  00 00 181E 1902 0003 00 00 07 06 01 O.K.
77  DISKFIX2.TXT  00 00 1EB2 1F0B 0028 00 00 07 07 01 O.K.
78  DISKFIX2.BIN  00 00 1018 1EB1 0005 00 00 07 07 01 O.K.
79  DISKFIX2.SYM  00 00 1F0C 1F0E 0003 00 00 07 07 01 O.K.
80  MINIOPS2.TXT  00 00 1A15 1B18 0022 00 00 07 06 01 O.K.

81  MINIOPS2.BIN  00 00 1819 1B1B 0003 00 00 07 06 01 O.K.
82  MINIOPS2.SYM  00 00 1B1C 1B1E 0003 00 00 07 06 01 O.K.
83  DISKFIX .CMD  00 00 2019 2102 0008 00 00 07 07 01 O.K.
84  DISKFMAN.TXT  00 00 3A14 3B0F 001A 00 00 07 06 01 O.K.
```

```
0010                    NAM     MINIOPS2 Block Read/Write Ops
0020                *                    5K version for DISKFIX
0030                *                    Rev 2.1  July 1, 1981
0040                    OPT     O,S,NOG

0060                *External References

0080       A97F     BUFEND EQU   $A97F    Last byte of FSB
0090       A880     FSB    EQU   $A880    System file sector buffer
0100       BE03     WRITEP EQU   $BE 3    Pointer to WRITE
0110       BE00     READP  EQU   $BE00    Pointer to READ
0120       BE06     VERIFP EQU   $BE06    Pointer to VERIFY
0130       BF50     DRVSL1 EQU   $BF50    2nd entry to DRVSEL
0140       BF39     RSTOR1 EQU   $BF39    2nd entry to RESTOR
0150       BF66     RDYCK1 EQU   $BF66    2nd entry to RDYCK
0160       BEF6     SEEK   EQU   $BEF6    Set track, sector & side
0170       E07E     POATA1 EQU   $E07E    Monitor string-output
0180       AD03     WARMS  EQU   $AD03    Reentry to FLEX
0190       E1AC     INEEE  EQU   $E1AC    Keyboard input routine

0210                *Page 0 Registers -- compatible with
0220                *MSI FDOS, which may be co-resident

0 0000                  ORG     $0000

0260 0000 0001  TRACK   RMB     1       Requested track #, 0-$4C
0270 0001 0001  SECTOR  RMB     1       Requested sector, 1-$1E
0280 0002 0001  WTRACK  RMB     1        orking location
0290 0003 0001  WSECT   RMB     1
0300 0004 0001          RMB     1       Not used
0310 0005 0001  TEMD    RMB     1       Used in MINIDOS
0320 0006 0002  CATENT  RMB     2       Catalog entry pointer
0330 0008 0001  BMEMH   RMB     1       Starting memory location
0340 0009 0001  BMEML   RMB     1       LSB
0350 000A 0001  EMEMH   RMB     1       Ending memory location
0360 000B 0001  EMEML   RMB     1       LSB
```

```
0037@ 000C 0001   NDSECT RMB   1        Block size in sectors
0038@ 000D 0001   STATUS RMB   1        Used by MSI only
0039@ 000E 0001   ERRWD  RMB   1        Code for error-type
0040@ 000F 0001   ERRCNT RMB   1        Failure/retry count
0041@ 0010 0001   RWWORD RMB   1        @ Verify, 1 Read, 2 Write
0042@ 0011 0001   WMEMH  RMB   1        Working location in memory
0043@ 0012 0001   WMEML  RMB   1        LSB
0044@ 0013 0001   ATRACK RMB   1        Head position (MSI only)
0045@ 0014 0004   ATRK@  RMB   4        Drive @-3 head pos (MSI only)
0046@ 0018 0001   DRIVE  RMB   1        Requested drive (MINIOPS)
0047@ 0019 0002   SAVES  RMB   2        SP Save
0048@ 001B 0002   PROGX  RMB   2        Return after error (MSI)
0049@ 001D 0001   SCOUNT RMB   1        Working sector-count
0050@ 001E 0002   PRDGX2 RMB   2        Return on error (MINIOPS)
00510             *Preset TRACK, SECTOR, NOSECT and EMEMH-L;
00520             *Enter MINIOPS with XR at memory location
00530             *for R/W, via selected pointer for R/W and
00540             *desired drive.

00560 5000                ORG   $5000

00580 5000 20 0D  MINIOP BRA    WARMP
00590 5002 01     V      FCB    1
00600 5003 7E 512D READ@ JMP    BREAD@   Read from Drive @
00610 5006 7E 5131 READ1 JMP    BREAD1   Read from Drive 1
00620 5009 7E 5067 WRITE@ JMP   BWRT@    Write to Drive @
00630 500C 7E 506B WRITE1 JMP   BWRT1    Write to Drive 1
00640 500F 7E AD03 WARMP JMP    WARMS    No proper entry here

00660             *MEMSET checks WMEMH-L, sets byte count in B

00680 5012 96 0A  MEMSET LDA A  EMEMH    Last data to go
00690 5014 D6 0B         LDA B  EMEML    LSB
00700 5016 00 12         SUB B  WMEML    Where we are
00710 5018 92 11         SBC A  WMEMH    Balance to be done
00720 501A 25 05         BCS    EXSET    Too far already?
00730 501C 26 02         BN     SETMEM   Lotsa room
00740 501E 5C            INC B           Last sector: add 1 byte.
00750 501F 39            RTS             1 to 256 bytes here.
00760 5020 5F     SETMEM CLR B           Count of "256"
00770 5021 39     EXSET  RTS             If C set, all done.

00790             *SECWRT loads the sector buffer, then
00800             *writes one sector to the FLEX disk

00820 5022 7F 000F SECWRT CLR   ERRCNT   Disk error flag
00830 5025 CE A800 SECWT@ LDX   @FSB     Start of buffer
00840 5028 0F            SEI
00850 5029 9F 19         STS    SAVES    Save SP
00860 502B 9E 11         LDS    WMEMH    Start of data block
00870 502D 34            DES             Will INS before PUL
00880 502E 32     LDBUF  PUL A           Data from memory
00890 502F A7 00  LDBUF1 STA A  0,X      Into FSB
00900 5031 08            INX
00910 5032 5A            DEC B
00920 5033 26 F9         BNE    LDBUF
00930 5035 4F            CLR A
00940 5036 5C            INC B          Keep falling through
00950 5037 8C A980       CPX    @BUFEND+1  Past end of buffer?
00960 503A 26 F3         BNE    LDBUF1   Fill with nulls
00970 503C 9E 19         LDS    SAVES
00980 503E CE A800 SECWT1 LDX   @FSB     Buffer address
00990 5041 96 02         LDA A  WTRACK
01000 5043 D6 03         LDA B  WSECT
01010 5045 BD BE83       JSR    WRITEP   Try & write
01020 5048 27 09         BEQ    WRET     Looks good.
01030 504A 8D 13         BSR    RESTD    Problems. Home.
01040 504C BD 50FE       JSR    INCERR   Step error count
01050 504F 25 ED         BCS    SECWT1   Not up to 5 yet?
01060 5051 16            TAB             5 is Write-error code
01070 5052 0D     WERR   SEC             Write-error
01080 5053 39     WRET   RTS

01100             *Fatal SEEK errors unchecked by FLEX in WRITE

01120 5054 96 02  SEEKT  LDA A  WTRACK   Working location
01130 5056 D6 03         LDA B  WSECT
01140 5058 BD BEF6       JSR    SEEK     Try and get there
01150 505B C4 10         AND B  #$10     Check for seek-error
01160 505D 27 F4         BEQ    WRET     No problem
01170 505F 8D 5128 RESTO JSR    DRSET    Drives ready?
01180 5062 25 EF         BCS    WRET     Some kind of error
01190 5064 7E 0F39       JMP    RSTDR1   Back to Track @

01210             *BWRITE routines handle block-write,
01220             *block-verify and sector-rewrite tries.

01240 5067 86 00  BWRT@  LDA A  #0       Drive @
01250 5069 20 02         BRA    BWRITE
01260 506B 86 01  BWRT1  LDA A  #1       Drive 1
01270 506D C6 02  BWRITE LDA B  #2       Write code
01280 506F BD 5106       JSR    DPREP@   Set up registers & drive
01290 5072 25 48         BCS    ERR      Not ready, zero count &c.
01300 5074 C4 40         AND B  #$40     Check for protected disk
01310 5076 26 44         BNE    ERR

01320 5078 8D DA         BSR    SEEKT    Assure correct track
01330 507A 25 40         BCS    ERR      Not ready or big problems
01340 507C 8D 94  BWRT2  BSR    MEMSET   Check WMEM, set ACC B
01350 507E 25 0F         BCS    ENDBW    All done?
01360 5080 8D A0         BSR    SECWRT   Write one sector
01370 5082 25 3B         BCS    ERR      Something is wrong
01380 5084 7A 001D       DEC    SCOUNT   One done
01390 5087 27 06         BEQ    ENDBW    Enough?
01400 5089 8D 56         BSR    SEC1NM   Step block & sector
01410 508B 25 2F         BCS    ERR      Past Track 77?
01420 508D 20 ED         BRA    BWRT2    Loop.
01430 508F DE 00  ENDBW  LDX    BMEMH    Back to beginnings
01440 5091 5F            CLR B           New RWWORD, @ (V)
01450 5092 D7 0F         STA B  ERRCNT   Clear for VERIFY
01460 5094 8D 74         BSR    DPREP1   Reset pointers
01470 5096 8D 5012 VERIF1 JSR   MEMSET   Get count
01480 5099 25 50         BCS    EXW      All done?
01490 509B 8D 87  VERIF2 BSR    SEEKT    Set track & sector
01500 509D 8D BE86 VERIF3 JSR   VERIFP   Do a verify (CRC)
01510 50A0 26 0E         BNE    F1X      Got a problem?
01520 50A2 7F 000F       CLR    ERRCNT   No problem
01530 50A5 7A 001D       DEC    SCOUNT   Sector count
01540 50A8 27 4E         BEQ    EXW      All done?
01550 50AA 8D 35         BSR    SEC1NM   Next block, next sector
01560 50AC 24 E8         BCC    VERIF1   Looks OK
01570 50AE 20 0C         BRA    ERR
01580 50B0 8D 4C  F1X    BSR    INCERR   Step ERRCNT
01590 50B2 24 0B         BCC    ERR      B has error-code
01600 50B4 8D 5012       JSR    MEMSET   Set byte-count in B
01610 50B7 8D 5025       JSR    SECWT@   Rewrite this one
01620 50BA 24 E1         BCC    VERIF3   Have another look now
01630 50BC D7 0E  ERR    STA B  ERRWD    Save code
01640 50BE CE 517A       LDX    #ERRTAB  Table of texts & codes
01650 50C1 F7 51EC       STA B  LMSG
01660 50C4 17            TBA
01670 50C5 A1 00  ERR1   CMP A  0,X      Check code
01680 50C7 27 0A         BEQ    ERRMSG   Matches? Run message.
01690 50C9 C6 04         LDA B  #4       EOT
01700 50CB 08     FNDE   INX
01710 50CC E1 00         CMP B  0,X      Look for end of msg.
01720 50CE 26 FB         BNE    FNDE     Loop
01730 50D0 08            INX             Step past EOT
01740 50D1 20 F2         BRA    ERR1
01750 50D3 08     ERRMSG INX             Past code
01760 50D4 BD E07E       JSR    PDATA1   Run text
01770 50D7 8E A07D       LDS    #$A07D   Reset stack pointer
01780 50DA BD E1AC       JSR    INEEE    Hold for keystroke
01790 50DD DE 1E         LDX    PROGX2   Designated return
01800 50DF 6E 00         JMP    0,X      Exit.

01820             *SEC1NM steps WSECT by 1 sector; WMEM by $100

01840 50E1 7C 0011 SEC1NM INC    WMEMH    Step to next block
01850 50E4 96 02   SEC1NC LDA A  WTRACK
01860 50E6 D6 03         LDA B  WSECT
01870 50E8 5C            INC B
01880 50E9 C1 1F         CMP B  #$1F
01890 50EB 25 07         BCS    SECSET
01900 50ED 4C            INC A
01910 50EE C6 01         LDA B  #1
01920 50F0 81 4D         CMP A  #$4D     Track 77
01930 50F2 24 06         BCC    SECERR   Oops!
01940 50F4 97 02   SECSET STA A  WTRACK
01950 50F6 D7 03         STA B  WSECT
01960 50F8 0C      EXW   CLC              All's well
01970 50F9 39            RTS
01980 50FA C6 FD   SECERR LDA B  #$FD
01990 50FC 0D            SEC
02000 50FD 39            RTS

02010 50FE 96 0F  INCERR LDA A  ERRCNT   Get error-count
02020 5100 4C            INC A          One more
02030 5101 97 0F         STA A  ERRCNT
02040 5103 81 05         CMP A  #5       C set, OK
02050 5105 39            RTS            Return, error code in B

02070             *DPREP initializes Page @ pointers

02090 5106 DF 00  DPREP@ STX    BMEMH    XR has S.A. in memory
02100 5108 97 18         STA A  DRIVE    Drive @ or 1
02110 510A D7 10  DPREP1 STA B  RWWORD   Read, write, verify
02120 510C DF 11         STX    WMEMH    Working location
02130 510E 96 01         LDA A  SECTOR
02140 5110 26 03         BNE    *+5      Not @, OK
02150 5112 7C 0001       INC    SECTOR   No "@" sector here
02160 5115 DE 00         LDX    TRACK    Requested track & sector
02170 5117 DF 02         STX    WTRACK
02180 5119 4F            CLR A
02190 511A 97 0F         STA A  ERRCNT
02200 511C 97 0E         STA A  ERRWD
02210 511E D6 0C         LDA B  NOSECT
02220 5120 07 1D         STA B  SCOUNT   Working count
02230 5122 26 04         BNE    DRSET    Not @? OK
02240 5124 C6 FE         LDA B  #$FE     Spec error
02250 5126 00            SEC
02260 5127 39            RTS
```

THE COMPLETE BUSINESS SYSTEM

+Multiuser+Highly Expandable+Cost Effective

S+ THE CONCEPT

The S+ system is a modular computer system in which all portions of the hardware and software are designed to work together in the most efficient way possible. An S+ single user system with floppy disk storage is a competitive and cost effective entry level system. Unlike most other small computers being sold as "personal", or "small business" machines, the S+ system may be expanded to maximum capabilities using this same hardware and software. You cannot end up with a DEAD END system that cannot be expanded and whose software is not compatible with larger machines. A basic S+ system may be expanded to thirty-two users, a megabyte of main memory and hundreds of megabytes of hard disk storage by simply plugging in, or connecting the desired upgrade equipment.

TOTAL DESIGN—Hardware and Software

The S+ system is an integrated hardware and software design. The two complement and enhance each other in this system. The UniFLEX® operating system used in the S+ systems is patterned after the Bell Laboratories UNIX® operating system, one of the most admired and widely used operating systems in the world. Instead of being an afterthought, the software is part of the design of the S+ system. You can be sure that with this approach that all parts of the computer operate with maximum efficiency and cost effectiveness.

THE CENTRAL PROCESSOR

The basic S+ system is configured with 256K bytes of memory and can be expanded to more than 1 million bytes. An efficient and fast hardware memory management system is used to allocate the available memory among the users on a dynamic basis. As little as 8K bytes, or the entire memory—if needed—can be used by any individual user. This makes it possible to run very large programs on the system, but it also uses no more memory than necessary for a particular job. The increase in cost effectiveness of this system over crude and outdated bank switching arrangements is dramatic.

The central processor runs in both user and supervisor states. It can detect and reject a defective user program. It is impossible for a user program to go bad and stop the entire system, as can happen quite easily in less sophisticated systems.

Task switching is accomplished by use of a multiple map RAM memory, with sixty-four individual task maps. Each task can access from 4 to 64 K-bytes of memory. Multiple tasks may be used in programs that require more than 64K bytes of memory for execution. When a task is completed the memory is automatically released for other use.

SOFTWARE

The S+ operating system, UniFLEX® is a multiuser, multitasking operating system based on the UNIX® operating system that has been used for many years on Digital Equipment Corp. PDP-11 series minicomputers. It is considered one of the most sophisticated and "user friendly" operating systems available. Variations of UNIX® are rapidly becoming standard on mini and larger microcomputers.

A large variety of languages are available for use with the system. These include FORTRAN, COBOL, BASIC, and Pascal. Word processing packages are also available to give you full text processing capability on the system.

Applications programs are available in large quantities in many fields. This includes general business, medical, dental, veterinary, library and real estate management; plus others. Since the system is multiuser it can also be connected to cash registers to produce a point-ofsale terminal system combined with the computer. The possibilities for application of this system are endless.

THE I/O SYSTEM

The S+ system is totally interrupt driven. All terminal and printer I/O devices connect to an I/O bus separate from the main bus. Up to thirty-two separate devices may be connected to the I/O bus at any one time. If I/O activity is great enough to cause an unacceptable slowdown in system operation, a separate I/O processor can be installed in the system. This plug-in option removes all I/O handling overhead from the main processor and allows operation of up to thirty-two external devices at 9,600 baud. Without an integrated total design, as in the S+ system, it would become impractical to use a UNIX®type operating system in a situation with heavy terminal I/O activity.

DISK STORAGE

A wide range of disk storage capacity is available for the S+ system, from 2.5 M-byte floppy disks to an 80 M-byte Winchester and many sizes between. All disk controllers use direct memory access (DMA) type operations to maximize data transfer and to minimize overhead on the main processor. The Winchester disks also use intelligent controllers along with DMA transfers to preserve the performance that these type devices are capable of giving. Without this distributed intelligence the system performance would be greatly degraded. The UniFLEX®operating system is designed to work at maximum efficiency with this type disk system. The data transfer rates achieved by this combination rival those of large minicomputers.

COMMUNICATIONS

A high speed local network communications system is available to interconnect S+ systems. The VIABUS® network will allow communication between systems at data rates of over 400K baud. Such a system makes it possible to share data between local systems in an efficient and low-cost manner.

AVAILABLE SOON

Tape backup—20M-Byte in less than 15 minutes on a standard ¼ inch cartridge.

Mini-Wini—5 and 10 M-Byte Winchesters—5¼ inch package. Winchester performance, for smaller systems in a small package. UniFLEX® compatible design.

Large Capacity—190 and 340 M-Byte Winchesters, plus SMD cartridge drives.

SOUTHWEST TECHNICAL PRODUCTS CORPORATION
219 W. RHAPSODY
SAN ANTONIO, TEXAS 78216 **(512) 344-0241**

```
0227Ø 512B 96 1B   DRSET  LDA A  DRIVE
0228Ø 512A 7E BF5Ø        JMP    DRVSL1   Drive set & ready-check

Ø23ØØ                *BREAD handles block-read operations

Ø232Ø 512D 86 ØØ   BREADØ LDA A  #Ø       Drive Ø
Ø233Ø 512F 2Ø Ø2           BRA    BREAD
Ø234Ø 5131 86 Ø1   BREAD1 LDA A  #1       Drive 1
Ø235Ø 5133 C6 Ø1   BREAD  LDA B  #1       Read operation
Ø236Ø 5135 8D CF           BSR    DPREPØ   Set up registers
Ø237Ø 5137 25 17           BCS    ERRP     Not ready or Ø sectors
Ø238Ø 5139 CE A88Ø BREAD2 LDX    #FSB     Read into sector buffer
Ø239Ø 513C 96 92           LDA A  WTRACK
Ø24ØØ 513E D6 93           LDA B  WSECT
Ø241Ø 514Ø BD BE8Ø         JSR    READP    Read one sector
Ø242Ø 5143 27 ØE           BEQ    TRANS    Looks OK
Ø243Ø 5145 BD 5Ø5F         JSR    RESTO    Home drive, fix registers
Ø244Ø 5148 25 Ø6           BCS    ERRP     Major problem
Ø245Ø 514A 8D B2           BSR    INCERR   Step error count
Ø246Ø 514C 25 EB           BCS    BREAD2   Not 5 yet
Ø247Ø 514E C6 FF           LDA B  #$FF     'Read error' code
Ø248Ø 5150 7E 508C ERRP    JMP    ERR      Give up

Ø2519 5153 9F 19   TRANS  STS    SAVES
Ø2520 5155 ØF              SEI             No interrupts!
Ø2530 5156 8E AB7F         LDS    #FSB-1   Will INS before a PUL
Ø2540 5159 DE 11           LDX    WMEMH
Ø2550 515B 5F              CLR B           Full count of 256
Ø2560 515C 32      TRANS1 PUL A
Ø2570 515D A7 ØØ           STA A  Ø,X      1 byte minimum transfer
Ø2580 515F 9C DA           CPX    EMEMH    At designated memory limit?
Ø2590 5161 27 14           BEQ    EXRD     If so, exit
Ø2600 5163 Ø8              INX
Ø2610 5164 5A              DEC B           Byte-count
Ø2620 5165 26 F5           BNE    TRANS1   Loop
Ø2630 5167 DF 11           STX    WMEMH    Advance working location
Ø2640 5169 9E 19           LDS    SAVES
Ø2650 516B 7A ØØ1D         DEC    SCOUNT   Sector-count
Ø2660 516E 2F Ø7           BLE    EXRD     All done?
Ø2670 5170 BD 50E4         JSR    SECINC   Next sector
Ø2680 5173 25 DB           BCS    ERRP     Off the end?
Ø2690 5175 20 C2           BRA    BREAD2   Loop for next sector
Ø2700 5177 9E 19   EXRD   LDS    SAVES    Recover SP
Ø2710 5179 39              RTS             Use it.


Ø2740                *ERRDR codes and messages


Ø2770 517A FF      ERRTAB FCB    $FF
Ø2780 517B 52              FCC    /Read error/
Ø2790 5185 Ø4              FCB    4
Ø2800 5186 FE              FCB    $FE
Ø2810 5187 43              FCC    /Count error/
Ø2820 5192 Ø4              FCB    4
Ø2830 5193 FD              FCB    $FD
Ø2840 5194 54              FCC    /Track overflow/
Ø2850 51A2 Ø4              FCB    4
Ø2860 51A3 Ø5              FCB    5
Ø2870 51A4 57              FCC    /Write error/
Ø2880 51AF Ø4              FCB    4
Ø2890 51BØ 8Ø              FCB    $8Ø
Ø2900 51B1 44              FCC    /Drive not ready./
Ø2910 51C1 Ø4              FCB    4
Ø2920 51C2 4Ø              FCB    $4Ø
Ø2930 51C3 5Ø              FCC    /Protected/
Ø2940 51CC Ø4              FCB    4
Ø2950 51CD 1Ø              FCB    $1Ø
Ø2960 51CE 52              FCC    /Record not found./
Ø2970 51DF Ø4              FCB    4
Ø2980 51EØ ØB              FCB    ØB
Ø2990 51E1 43              FCC    /CRC error./
Ø3ØØØ 51EB Ø4              FCB    4
Ø3Ø1Ø 51EC ØØ      LMSG   FCB    Ø
Ø3Ø2Ø 51ED 44              FCC    /Disk error - uncl./
Ø3Ø3Ø 51FF Ø4              FCB    4

Ø3Ø5Ø                      END    BMEMH  ØØØ8
BUFEND A97F                       BMEML  ØØØ9
FSB    A88Ø                       EMEMH  ØØØA
WRITEP BE83                       EMEML  ØØØB
READP  BE8Ø                       NOSECT ØØØC
VERIFP BE86                       STATUS ØØØD
DRVSL1 BF5Ø                       ERRWO  ØØØE
RSTOR1 BF39                       ERRCNT ØØØF
ROYCK1 BF66                       RWWORD ØØ1Ø
SEEK   BEF6                       WMEMH  ØØ11
PDATA1 EØ7E                       WMEML  ØØ12
WARMS  AØ93                       ATRACK ØØ13
INEEE  E1AC                       ATRKØ  ØØ14
TRACK  ØØØØ                       DRIVE  ØØ18
SECTOR ØØØ1                       SAVES  ØØ19
WTRACK ØØØ2                       PROGX  ØØ1B
WSECT  ØØØ3                       SCOUNT ØØ1D
TMD    ØØØ5                       PROGX2 ØØ1E
CATENT ØØØ6                       MINIDP 5ØØØ
V      5ØØ2
READØ  5ØØ3
READ1  5ØØ6
WRITEØ 5ØØ9
WRITE1 5ØØC
WARMP  5ØØF
MEMSET 5Ø12
SETMEM 5Ø2Ø
EXSET  5Ø21
SECWRT 5Ø22
SECWTØ 5Ø25
LDBUF  5Ø2E
LDBUF1 5Ø2F
SECWT1 5Ø3E
WERR   5Ø52
WRET   5Ø53
SEEKT  5Ø54
RESTO  5Ø5F
BWRTØ  5Ø67
BWRT1  5Ø68
BWRITE 5Ø6D
BWRT2  5Ø7C
ENDBW  5Ø8F
VERIF1 5Ø96
VERIF2 5Ø9B
VERIF3 5Ø9D
F1X    5ØBØ
ERR    5ØBC
ERR1   5ØC5
FNDE   5ØCB
ERRMSG 5ØD3
SEC1NM 5ØE1
SEC1NC 5ØE4
SECSET 5ØF4
EXW    5ØF8
SECERR 5ØFA
1NCERR 5ØFE
DPREPØ 51Ø6
DPREP1 51ØA
DRSET  512Ø
BREADØ 512D
BREAD1 5131
BREAD  5133
BREAD2 5139
ERRP   515Ø
TRANS  5153
TRANS1 515C
EXRD   5177
ERRTAØ 517A
LMSG   51EC

TOTAL ERRORS ØØØØØ


ØØØ1Ø                NAM    DISKFIX1 Check & Repair Crashed Disk
ØØØ2Ø                *      Rev 9.8  July 7, 1981
ØØØ3Ø                *By     Geoffrey A. Gass
ØØØ4Ø                *      Portland, Oregon 97201

ØØØ6Ø                       OPT    O,NOG

ØØØ8Ø                *External References

ØØ1ØØ      ØØØØ     TRACK  EQU    Ø
ØØ11Ø      ØØØ5     TEMD   EQU    5
ØØ12Ø      ØØØ6     CATENT EQU    6
ØØ13Ø      ØØØA     EMEMH  EQU    $A
ØØ14Ø      ØØØC     NOSECT EQU    $C
ØØ15Ø      ØØ18     DRIVE  EQU    $18
ØØ16Ø      ØØ22     SAVEX  EQU    $22
ØØ17Ø      ØØ1E     PROGX2 EQU    $1E
ØØ18Ø      5133     BREAD  EQU    $5133
ØØ19Ø      5Ø6D     BWRITE EQU    $5Ø6D
ØØ2ØØ      A88Ø     SFSB   EQU    $A88Ø
ØØ21Ø      ACØ2     EOLCH  EQU    $ACØ2    End-of-line character
ØØ22Ø      ACØ3     DEPTH  EQU    $ACØ3    Page length
ØØ23Ø      ACØ9     PAUSE  EQU    $ACØ9    Halt at end of page
ØØ24Ø      AC11     LTERM  EQU    $AC11    Last delimiter
ØØ25Ø      AC14     LBUFP  EQU    $AC14    Line buffer pointer
ØØ26Ø      ADØ3     WARMS  EQU    $ADØ3    FLEX reentry
ØØ27Ø      ADØF     OUTCH  EQU    $ADØF    Selected output routine
ØØ28Ø      AD1B     PUTCHR EQU    $AD1B
ØØ29Ø      AD1B     INBUFF EQU    $AD1B    Buffer input
ØØ3ØØ      AD1E     PSTRNG EQU    $AD1E    String output with CRLF
ØØ31Ø      AEA9     PSTRG1 EQU    $AEA9    String output, no CRLF
ØØ32Ø      AD24     PCRLF  EQU    $AD24
ØØ33Ø      AD36     ADOBX  EQU    $AD36
ØØ34Ø      AD39     OUTDEC EQU    $AD39
ØØ35Ø      AD3C     OUTHEX EQU    $AD3C
ØØ36Ø      AD45     OUTADR EQU    $AD45
ØØ37Ø      AD4B     INDEC  EQU    $AD4B
ØØ38Ø      BØØE     OUTL   EQU    $BØØE    Output left nybble in A
ØØ39Ø      8Ø12     OUTR   EQU    $8Ø12    Output right nybble in A
ØØ4ØØ      E87E     PDATA1 EQU    $E87E    String output per XR
ØØ41Ø      5659     TITLET EQU    $5659
ØØ42Ø      566E     DRVT   EQU    $566E
ØØ43Ø      5681     LOADTX EQU    $5681
ØØ44Ø      568E     NDTLT  EQU    $568E
ØØ45Ø      5698     LINKTX EQU    $5698
ØØ46Ø      567E     NBOOTX EQU    $567E
ØØ47Ø      56A7     MARTX  EQU    $56A7    Control sequence string
ØØ48Ø      56AE     EPAREN EQU    $56AE
ØØ49Ø      55ØB     FREECK EQU    $55ØB
ØØ5ØØ      5461     TRACE  EQU    $5461
ØØ51Ø                *DISKFIX traces directory entries, maps used
ØØ52Ø                *sectors, restores free-chain linkages, etc.

ØØ54Ø 52ØØ                  ORG    $52ØØ

ØØ56Ø 52ØØ 2Ø ØC   DISKFX BRA    DISKF1
ØØ57Ø 5202 Ø1      VER    FCB    1        Version 1
ØØ58Ø 5203 ØØØ2    CATSEC RMB    2        Disk addr, current entry
ØØ59Ø 5205 ØØØ2    DIRNUM RMB    2        Number of current entry
ØØ6ØØ 5207 ØØØ2    DISKAD RMB    2        Working disk addr (T & S)
ØØ61Ø 5209 ØØØ2    ADRLIM RMB    2        Disk addr upper limit
ØØ62Ø 52ØB ØØØ1    PGMERR RMB    1        Files w/ link/count errors
ØØ63Ø 52ØC ØØØ1    FREERR RMB    1        Errors in free chain
ØØ64Ø 52ØD ØØØ1    ERRC   RMB    1        Working error count

ØØ66Ø                *DISKF1 gets Drive #, sets Page length
```

```
00680 520E CE 5659 DIS F1 LDX   #TITLET  Clear screen, title
00690 5211 BD E07E        JSR   PDATA1   to terminal only
00700 5214 CE A003        LDX   #WARMS   Re-entry to FLEX
00710 5217 DF 1E          STX   PROGX2   Return from errors
00720 5219 B6 AC11        LDA A LTERM    Last delimiter
00730 521C B1 00           MP A #$00     C/R?
00740 521E 27 05          BEQ   DIS F2   Ask drive
00750 5220 B1 AC02        CMP A EOLCH    Special terminator?
00760 5223 26 09          BNE   DRVSL
00770 5225 CE 566E DIS F2 LDX   #DRVT    "Which drive? "
00780 5228 BD E07E        JSR   PDATA1   Run query
00790 522B BD AD1B        JSR   INBUFF   Get response
00800 522E BD AD4B DRVSL  JSR   INDEC    Get buffer data
00810 5231 25 F2          BCS   DIS F2   Nonsense?
00820 5233 5D             TST B          Any input at all?
00830 5234 27 EF          BEQ   DIS F2
00840 5236 DF 22          STX   SAVEX    4-digit input
00850 5238 96 23          LDA A SAVEX+1  LSB
00860 523A 81 03          CMP A #3
00870 523C 22 E7          BHI   DISKF2   Anything higher, NG
00880 523E 97 18          STA A DRIVE    Page 0 reference
00890 5240 86 AD10        LDA A OUTCH+1  Check output address
00900 5243 81 AC          CMP A #$AC     If $ACE4, "P" is on
00910 5245 26 0E          BNE   CLRM     Not to printer. Skip next.
00920 5247 86 30          LDA A #$30     48-line page
00930 5249 B7 AC03        STA A DEPTH    Set page-length
00940 524C B7 AC09        STA A PAUSE    Enable PAUSE feature.
00950 524F CE 56A7        LDX   #MARTX   Margin set for Diablo
00960 5252 BD AEA9        JSR   PSTRG1   Run control string

01010             *Clear mapping area

01030 5255 CE 5800 CLRM   LDX   #$5800
01040 5258 6F 00   CLRM1  CLR   0,X
01050 525A 08             INX
01060 525B 8C 7000        CPX   #$7000
01070 525E 26 F8          BNE   CLRM1
01080 5260 CE 0000        LDX   #0       Two bytes of nothin'
01090 5263 FF 520C        STX   FREERR   Also clears ERR

01110             *Get system record from disk

01130 5266 CE 0003        LDX   #3       Track 0 Sector 3
01140 5269 DF 00          STX   TRACK
01150 5268 CE A980        LDX   #$A980
01160 526E DF 0A          STX   EMEMH    Limit for entire program
01170 5270 CE 6E00        LDX   #$6E00   Where system record goes
01180 5273 BD 533B        JSR   DREAD    Get 1 sector
01190 5276 CE 6E10        LDX   #$6E10   Where data start
01200 5279 A6 16          LDA A $16,X    Last physical track
01210 527B E6 17          LDA B $17,X    Sector
01220 527D 5C             INC B          Allow use of last sector
01230 527E B7 5209        STA A ADRL1M   Starting limit
01240 5281 F7 520A        STA B ADRL1M+1 Used in Map Test
01250 5284 BD AD24        JSR   PCRLF
01260 5287 C6 08          LDA B #$8      Count for disk name
01270 5289 8D 5C          BSR   CONTO    Output disk name
01280 528B 8D 7A          BSR   OUTS1    And a space
01290 528D 86 23          LDA A #$23     "#"
01300 528F 8D 78          BSR   OUTEP
01310 5291 5F             CLR B
01320 5292 8D 64          BSR   DECOU    Volume number
01330 5294 8D 60          BSR   ADDOS    Start of free chain
01340 5296 8D 68          BSR   ADDOS    End of free chain
01350 5298 C6 FF          LDA B #$FF     Set leading spaces
01360 529A 8D 5C          BSR   DECOUT   Sector count
01370 529C BD 5335        JSR   DATO1    Creation date
01380 529F 8D 66          BSR   OUTS1
01390 52A1 8D 60          BSR   ADDOS    Last physical T & S
01400 52A3 8D AD24        JSR   PCRLF
01410 52A6 8D AD24        JSR   PCRLF

01430             *Check Boot Loader on disk

01450 52A9 CE 0001        LOX   #1       Track 0 Sector 1
01460 52AC 0F 00          STX   TRACK
01470 52AE CE A800        LDX   #SFSB    File sector buffer
01480 52B1 BD 533B        JSR   DREAD    Leaves data in buffer
01490 52B4 CE A800        LOX   #SFSB
01500 52B7 E6 05          LDA B 5,X      Link byte location
01510 52B9 EA 06          ORA B 6,X      2nd link byte
01520 52BB EE 00          LDX   0,X      First check for loader
01530 52BD 8C 8EA0        CPX   #$8EA0   1st instr is LDS #$A07D
01540 52C0 26 19          BNE   NBOOT    If not, no loader.
01550 52C2 CE 5681        LDX   #LOADTX  "Boot Loader" text
01560 52C5 8D AEA9        JSR   PSTRG1   Run it
01570 52C8 CE 568E        LDX   #NOTLT   " not linked.*
01580 52CB 5D             TST B          Check it
01590 52CC 27 10          BEQ   BOOT2    Run text
01600 52CE CE 5698        LDX   #LINKTX  " linked to "
01610 52D1 8D AEA9        JSR   PSTRG1
01620 52D4 CE A805        LDX   #SFSB+5  Point to link
01630 52D7 8D 2A          BSR   ADDOS    Run track & sector
01640 52D9 20 06          BRA   DIRTP    Check directory

01660 52DB CE 567E NBOOT  LDX   #NBOOTX  "No Boot Loader"

01670 52DE BD AEA9 BOOT2  JSR   PSTRG1
01680 52E1 BD AD24 DIRTP  JSR   PCRLF
01690 52E4 7E 5378        JMP   DIRTST   Check out directory

01710             *CONTO outputs (B) characters from string, per XR

01730 52E7 A6 00   CONTO  LDA A 0,X
01740 52E9 2B 04          BMI   CONTO1   Skip flagged character
01750 52EB 81 20          CMP A #$20     Check for non-printing
01760 52ED 24 02          BCC   *+4      Looks OK
01770 52EF 86 20   CONTO1 LDA A #$20     Substitute space
01780 52F1 BD 16          BSR   OUTEP
01790 52F3 08             INX
01800 52F4 5A             DEC B
01810 52F5 2E F0          BGT   CONTO    Loop
01820 52F7 39             RTS

01840             *DECOUT tidies up for FLEX OUTDEC routine

01860 52F8 DF 22   DECOUT STX   SAVEX    XR clobbered by OUTDEC
01870 52FA BD AD39        JSR   OUTDEC   Convert 16-bit BIN to DEC
01880 52FD DE 22          LDX   SAVEX
01890 52FF BD 05          BSR   OUTS     INX & output space
01900 5301 20 03          BRA   OUTS     One more time

01920             *ADDOS outputs 4-digit hex number and space

01940 5303 BD AD45 ADDOS  JSR   OUTADR   FLEX routine
01950 5306 08      OUTS   INX            Omitted in FLEX
01960 5307 86 20   OUTS1  LDA A #$20     Space
01970 5309 BD AD1B OUTEP  JSR   PUTCHR   Output
01980 530C 0C             CLC            No flags
01990 530D 39             RTS

02010             *HEXOS outputs 2 hex digits and space

02030 530E BD AD3  HEXOS  JSR   OUTHEX   FLEX routine
02040 5311 20 F3          BRA   OUTS     INX & output space

02060 5313 BD 02   SP6    BSR   SP2
02070 5315 BD 00   SP4    BSR   SP2
02080 5317 BD EE   SP2    BSR   OUTS1
02090 5319 20 EC          BRA   OUTS1

02110             *DECOS converts one byte to 2 decimal digits 0-99

02130 531B 5F      DECOS  CLR B          Tens counter
02140 531C A6 00          LDA A 0,X      Get byte
02150 531E 81 0A   DECOS1 CMP A #$A      Check for 0-9
02160 5320 25 06          BCS   SUM      Balance below 10
02170 5322 CB 10          ADD B #$10
02180 5324 80 0A          SUB A #$A
02190 5326 20 F6          BRA   DECOS1
02200 5328 1B      SUM    ABA            Assemble BCD
02210 5329 16             TAB            Save for LSB
02220 532A BD B00E        JSR   OUTL     Part of OUTHEX (FLEX)
02230 532D 17             TBA
02240 532E BD B012        JSR   OUTR     Now LSB
02250 5331 20 D3          BRA   OUTS     And space.

02270 5333 8D D9   DATO   BSR   HEXOS    Revision #(MINI file)
02280 5335 8D E4   DATO1  BSR   DECOS    File date - MO
02290 5337 BD E2          BSR   DECOS    DA
02300 5339 20 E0          BRA   DECOS    YR

02320             *DREAD reads 1 sector to memory at (XR)

02340 533B 86 01   DREAD  LDA A #1
02350 533D 97 0C          STA A NOSECT
02360 533F 96 18          LDA A DRIVE    Preset at outset
02370 5341 7E 5133        JMP   BREAD    MINIOPS routine

02390             *DWRITE writes $100 bytes to disk per XR

02410 5344 86 01   DWRITE LDA A #1
02420 5346 97 0C          STA A NOSECT
02430 5348 96 18          LDA A DRIVE    Constant preset
02440 534A 7E 5060        JMP   BWRITE   Track & sector were preset

02460             *DISPC displays 1 catalog entry

02480 534D A6 08   DISPC  LDA A $8,X     File type
02490 534F B5 01          BIT A #1       Check for program file
02500 5351 26 01          BNE   *+3      Got one?
02510 5353 4F             CLR A          00 flags FLEX or DATA file
02520 5354 97 05          STA A TEMD     To steer choice below
02530 5356 C6 08          LDA B #8       File name length
02540 5358 BD BD          BSR   CONTO    Display it
02550 535A 86 2E          LDA A #$2E     Separator
02560 535C BD A8          BSR   OUTEP
02570 535E C6 03          LDA B #3       Extension
02580 5360 BD 85          BSR   CONTO
02590 5362 BD 83          BSR   SP2      Couple spaces
02600 5364 BD A8          BSR   HEXOS    Attributes (hex)
02610 5366 BD A6          BSR   HEXOS    Last-sector byte-count
02620 5368 BD 99          BSR   ADDOS    1st track & sector
```

```
02630 536A 8D 97          BSR    ADDOS    Last track & sector
02640 536C 8D 95          BSR    ADDOS    File length
02650 536E 8D 9C          BSR    HEXOS    File sector map
02660 5370 D6 05          LDA B  TEMD     FLEX or MINI file?
02670 5372 27 BF          BEQ    DATO     FLEX, or MINI data file
02680 5374 8D 8D          BSR    ADDOS    MINI program file: SA
02690 5376 20 88          BRA    ADDOS    Entry address.

02710             *Directory Tests

02730 5378 4F       DIRTST CLR A
02740 5379 B7 5205         STA A  DIRNUM   Will assign numbers to
02750 537C B7 5206         STA A  DIRNUM+1 all files. Free = "0".
02760 537F CE 0005         LDX    #5       Track 0, Sector 5
02770 5382 FF 5203 DIRT1   STX    CATSEC   Directory T&S pointer
02780 5385 DF 00           STX    TRACK
02790 5387 BD AD24         JSR    PCRLF
02800 538A CE 6F00         LDX    #$6F00   Buffer for data
02810 538D BD AC           BSR    DREAD    Read 1 sector
02820 538F CE 6F10         LDX    #$6F10   Where data start
02830 5392 DF 06   DIRT2   STX    CATENT   Working location
02840 5394 A6 00           LDA A  0,X      Check 1st byte
02850 5396 27 57           BEQ    NEXD1    00? Blank. Pursue to end.
02860 5398 2A 12           BPL    DIRT3    Normal entry
02870 539A BD 5313         JSR    SP6      Deleted file - no number
02880 539D 86 28           LDA A  #'(      Put name in parentheses
02890 539F BD 5309         JSR    OUTEP    Output parenthesis
02900 53A2 8D A9           BSR    DISPC    Display data
02910 53A4 CE 56AE         LOX    #EPAREN  " -- Deleted)"
02920 53A7 BD AEA9         JSR    PSTRG1
02930 53AA 20 40           BRA    NEXDIR   Step to next entry
02940 53AC A6 08   DIRT3   LDA A  $08,X     Check attributes byte
02950 53AE B1 0F           CMP A  #$DF
02960 53B0 25 21           BCS    DIRT4    Not $E0 or $E1, readable
02970 53B2 BD 5313         JSR    SP6      Will not have a number
02980 53B5 BD 5307         JSR    OUTS1    Get name in line
02990 53B8 A6 0D           LDA A  $D,X     Start track
03000 53BA E6 0E           LDA B  $E,X     Sector
03010 53BC B1 5209         CMP A  ADRLIM   MSB
03020 53BF 25 07           BCS    LIMSET   New lower track
03030 53C1 26 08           BNE    DIRT31   Above ADRLIM
03040 53C3 F1 520A         CMP B  ADRLIM+1
03050 53C6 24 06           BCC    DIRT31   At or above limit

03070 53C8 B7 5209 LIMSET  STA A  ADRLIM   Set new upper limit
03080 53CB F7 520A         STA B  ADRLIM+1

03100 53CE BD 534D DIRT31  JSR    DISPC    Display, no comment
03110 53D1 20 19           BRA    NEXDIR

03130 53D3 FE 5205 DIRT4   LDX    DIRNUM   Generate next file number
03140 53D6 08              INX
03150 53D7 FF 5205         STX    DIRNUM
03160 53DA CE 5205         LDX    #DIRNUM
03170 53DD C6 01           LDA B  #1       Spaces for leading zero's
03180 53DF BD AD39         JSR    OUTDEC
03190 53E2 BD 5317         JSR    SP2
03200 53E5 DE 06           LDX    CATENT
03210 53E7 BD 534D         JSR    DISPC    Display data
03220 53EA 8D 75           BSR    TRACE

03240 53EC BD AD24 NEXDIR  JSR    PCRLF
03250 53EF DE 06   NEXD1   LDX    CATENT
03260 53F1 C6 18           LDA B  #$18     Interval to next entry
03270 53F3 BD AD36         JSR    ADDBX
03280 53F6 8C 7000         CPX    #$7000   To end of buffer?
03290 53F9 26 97           BNE    DIRT2    If not, read next entry
03300 53FB FE 5203         LDX    CATSEC   This block
03310 53FE 08              INX              Next sector
03320 53FF 8C 001F         CPX    #31F     To 001F?
03330 5402 27 03           BEQ    NEXD2    If so, all done
03340 5404 7E 5382         JMP    DIRT1    Get new sector
03350 5407 86 520D NEXD2   LDA A  ERRC     Error count
03360 540A B7 5208         STA A  PGMERR   Count and link errors
03370 540D 7E 550B         JMP    FREECK   Check free sectors now

03390                      END
```

DISKFIX2 to be concluded next month...

BIT Bucket

Dear Mr Williams Sr. :

I am deeply grateful to you and your magazine! I recently purchased a two drive 5.25" disk system with MPI 51 drives. Without your magazine, and the articles in it concerning disk systems, FLEX, MPI drives, and associated modifications. I would have had an extremely hard time getting the disk system going. As it was, I had no trouble at all!

I even have FLEX formatting for 40 tracks and stepping at a 12ms rate! Thanks again for your outstanding magazine!

I subscribe to five other computer magazines, but '68' Micro Journal is the only one that I eagerly await, and read from cover to cover (INCLUDING advertisements!) so that I can be kept current with the developements in the 68xx field.

I especially enjoy, and receive much information from, Ronald W. Anderson's series "FLEX User Notes".

Keep up the good work and best wishes in the coming years!

Sincerely yours,

Jerr P. Starzinski
POB 9456
Yakima, WA 98909

ED COLLE & ASSOCIATES
DESIGNERS OF DIGITAL EQUIPMENT
322-A BREESPORT
SAN ANTONIO TEX. 78216
PH. (512) 340-3957

DON WILLIAMS
5900 CASSANDRA-SMITH
HICKSON TENN. 37343

Dear Don,

The MARKSMAN ROM to update the performance of your Hard-Disk is enclosed, along with a much improved format program and disk repair utility. This is the last ROM that I wrote for the MARKSMAN and is far better than the MARK 2 ROM that is presently being used with the MARKSMAN for FLEX operating systems.

The version of the MARKSMAN controller presently being sold by SWTPC was designed in 1977 and operational (with a CALIUS Hard-Disk) in 1977. It was scheduled to go into production in early 1978. But the announcement of the MARKSMAN drives caused them to cancel the project until the availability of the MARKSMAN. The delivery problems with the MARKSMAN drives prevented delivery of the first units until Sept. 1978. Other factors delaying the introduction of the hard-disk were the switching to FLEX operating system from DOS and the introduction of the 6809.

In response to your question, concerning use of multiple drives with the controller, the answer is yes and no. FLEX only supports 4 drives and SWTPC decided that two must be floppies. This was to give limited backup capability, a feature lost in the switch from the CALIUS with one fixed and one removable platter. The controller has all of the electronics to run 16 drives.

I never did a ROM for the multiple drives, nor did I complete the board for the input buffers to go on the slave drives. The reason: FLEX is not fast enough nor versatile enough to support more than one hard-disk. The introduction of UNIFLEX will solve these problems as the system matures, and additional features are added to it.

I have included three sets of drivers for your use, called HDD, HDI and HD2.

HDD is best at 1 MC.
HDI is best at 1.8 and 2 MC.
HD2 is a two controller driver, with one controller set as drive 2 at $F100 and the other as drive 3 at $F300.

Several dealers are using the HD2 as a backup device on systems. That is, they use two MARKSMANS, one to back up the other. This allows complete backup in less than an hour. It took me 6 hours yesterday to back up my system on 8" floppies. I have approximately 900 files, taking up 10 megabytes.

All of the drivers allow the use of minifloppies and do not care if 5" or 8" drives are used. My system has 8", 8", and hard-disks.

About the speed of the ROM -- you will notice a drastic difference from the original ROM that you are using. The Mark 2 ROM (the current FLEX ROM) corrected several problems in the Disk ROM (written by SWTPC). It added a work queue, allowing the Disc to log 7 operations behind the computer, and look ahead, allowing it to work ahead of the operating system. The system is no longer I/O bound. The Disk can return at least five sectors in the 26 millisecond rotation of the Disc at 2 MC. FLEX can, in normal operation, handle only 2 or 3.

The 8 hour format that you mentioned must have been the original SWTPC format program. I thought it was faster than that, however. Watch the new format program. The first pass takes 70 seconds to write 81,244 sectors. It writes headers and valid sector data. The second pass puts links in each sector (7 minutes) at 170 sectors per second. The third pass (also 7 minutes) checks each sector for correct links and errors. This format program could be speeded up by 80 percent, to allow the format to be done in eight minutes, if excruciatingly necessary.

This ROM is called MARK 3 and overcomes the limitations on error correction and recovery, caused by the need to use a single IOE for MARK 2. It uses a 32K EPROM and also has provisions to double the buffer size to 16 disk operations. This may help random files to run faster -- I'll let you know.

I have a better version of the controller, but it may never see production due to the 8" disks cutting it out, just as the HARKSMAN cut out the cartridge type drives. The new controller stacks up 35 operations in its work queue with a typical 4k task swap taking 2 milliseconds of computer time, with only one interupt to service.

I'll provide the EPROM and software to any of your customers for a cost of $100. These versions are not available through SWTPC. If there is a demand, I will also provide a free space sort and error history handling file. The error history file would automatically update a bit error map, to be used by the format program to screen the disk for marginal sectors. Sectors known to be bad would then be automatically linked out. It is very difficult to find a bad sector without testing it for thousands of cycles; this is impractical except on suspected sectors. Hard-disk errors in most cases do not show up until several days after the data is written (soft errors becoming hard errors).

ED COLLE

FOUND - A GIMIX USER IN ANTARCTICA

We received a letter from William Dickinson of Windrush Micro Designs Ltd. in England stating -

"last but not least we suggest that you amend your advertisement. The two parallel interface boards on this order will be integrated into a system used by the British Antarctic Survey for data logging in ..... you guessed it, the Antarctica."

Don Williams, Editor
Computer Publishing, Inc.
'68 Micro Journal
5980 Cassandra Smith
Hixson, TN. 37343

Dear Don:

I have been using MPI B-52 drives in my own and several customers' systems for about two years. Until recently, I have been quite pleased with their reliability and speed. Recently, however, several of the drives have suffered failures of the main bearings. Investigating the problem produced some results which should be of interest to users of 5" and 8" floppy drives.

In order to save time waiting for motor-on delays, many disk controller boards provide a timer which causes the drive motors to remain on for some period after the last access. The theory is that most disk accesses will occur within a short period of each other and that the motors should remain on for the duration. Unfortunately, many disk controller boards have a timer period which causes an excessively-long motor-on time. In many business applications, this may cause the motors to remain on for very long periods, even up to several hours per day. Over a period of time, this causes excessive media, motor, and bearing wear.

The obvious solution is to reduce the motor-on time. The method required to do this varies across systems and controller boards, of course. However, a large number of the controller boards (such as SWTPC DC3, DC4, and CHAP2 and SMS DDC16) use a 14541 to control the motor-on timing. In this case, the resistor connected to pin 1 of the 14541 may be reduced in value to shorten the delay. According to the data sheet, the lowest permissable value of this resistor is 10K ohms. I have shortened the delays on several of my disk controller boards to about 15 seconds, which seems to work quite well.

Sincerely,
Bud Pass
Dr. E. M. (Bud) Pass

Don,

I am still going to school and I will be finished next spring. I am holding 3.25 GPA in my computer classes. I have not had much time to play with my SWTPC 6800 but I try when I have time. I am taking four computer classes which is considered a full load.

I would like to point out to your readers a very good product I have purchased this year. It is the DNA VIDEO ADAPTER FOR YOUR TERMINAL by Johnson Micro Computer. It works great, installed easily and I highly recommend it to anyone who gets tired of 1200 baud on the CT-64. It can also be used for timesharing just as if you had the regular memory board in the terminal plugged in. Publish this letter if you can. Thanks for the good magazine.

Thomas J. Mattingly
1005 Essex Dr. West
Las Vegas, NV 89107
Tom Mattingly

Larry Williams,
Executive Editor,
'68' Micro Journal,
P.O. Box 849,
NIXSON TN 37343
U.S.A.

Roger D. Knowles
24 Junction Street,
RINGWOOD VICTORIA 3134
AUSTRALIA

Dear Larry,

This letter is to confirm the following facts which I related to you by telephone, and to request the assistance of yourself or your readers:-

(a) I ordered a DMB 64-256 memory board from Southwest Microsystems of Austin, Texas, on 23 May 1981 after seeing their advertisement in the Journal.

(b) A board was supplied to me on 12 June 1981. The board was labelled DMB 64-256 Rev A, does not have the same component layout as that advertised, is not silk screened with component legends, does not include an advertised RAM defeat and more importantly, I have find out, does not work in its' present state.

(c) Prior to incorporating this board into my system, I telephoned Southwest Microsystems and requested that they forward the 'full documentation' advertised, which had not been provided. I was promised that it would be sent, together with the unused portion of postage charges which I had provided. That was a month ago, and neither has been received.

(d) Subsequent telephone calls have been unanswered, and on 21 August 1981 I got a 'number not connected' recording from the Telephone Company, and they now advise that they have no listing for this Company.

(e) On incorporating this board into my 6800 system within the last few days, I find that it won't function normally, 'bloomps' back into the system and eventually hangs it up.

If you of any of your readers have a copy of the documentation, including (hopefully) schematics, I would very much appreciate a copy, for which I would pay, so that I can try to work my way out of this mess. It looks like costing me $41) plus phone calls plus a new memory board at the moment.

Yours faithfully,

Roger D. Knowles

26 August 1981

Ed's Note: If any of you have any information that may assist Roger, and other readers, concerning this memory board or what you might have done to get one to run properly, please let me know.

We of course halted their advertising and alerted our readers a few months back, however, some who are having trouble would appreciate anything you might have done. If you will send it to me I will see that it is published. Hopefully most if not all can salvage their boards.

We have attempted to contact them at the following address and telephone to no avail:
Southwest Microsystems, 6803 Kings Point Lane, Austin, TX 78723,
Attention Mr. William Gordon, (615) 928-2162, 928-1202.

Also no reply to our 'registered mail' or telephone calls. It appears they have ceased business and have had the telephone service disconnected. I try hard to keep this type thing from happening, we have tightened up our review and approval for advertising policies as a result of these sort of poor business practices. Still a bad apple will slip in and I will NOT ALLOW ANY ADVERTISING of a product that does not function as advertised, I am not a 'police agency' for the Standard S50 Bus group, or any other group. However, I can still control what is advertised in 68 Micro Journal and I will do all I can to see to it that the advertising you read here is accurate.

I know that our advertising policies are far, far more strict than for any other magazine I know of. Yet it can (and has) happened to us, but not if I know about it in time. That is where you come in, I HAVE TO KNOW AND IN WRITING! Together we can keep this sort of thing from occurring or at least hold it to a bare minimum.

DMW - - -

TEXAS COMPUTER

817-275-1848 • P.O. BOX 120816 • ARLINGTON, TX 76012

October 10,1981

Mr. Don Williams
'68 Micro Journal
5900 Casamira Smith
Nixson,Tennessee 37343

Dear Mr. Williams:

TEXAS COMPUTER would like to announce the immediate availability of the TC-1 bit rate generator board for the 5S50 bus.

The board provides 9 standard bit rates from 19.2k to 110 simultaneously. Any of the bit rates are available at a jumper area and the user need only jumper the appropriate rate for buffering onto the bus.

The bit rates are crystal controlled and allow the user to run a terminal at 19.2k while running another device at 110 baud.

MPA owners please note that this board requires no upgrade kit. Modifications are extremely simple and documentation and instructions are included.

NO MC14411 chip is used on this board and all IC's are sockated.

The TC-1 board is available as a tested, complete board with gold connectors standard for $49.00 + $3.00 for shipping. Texas residents please add 5% for sales tax.

Please see our ad elsewhere in this issue!

Sincerely,

Dave C. Bolen

P.S. WATCH FOR OUR DUAL PORT CARD---COMING SOON!!

R. ZEFF
ZEFF ADVANCED PRODUCTS
2135 STONE AVE.
MODESTO, CA. 95351

'68' MICRO JOURNAL
1918 HAMILL RD.
P.O. BOX 849
NIXSON, TN 37343

The following program was written to enable a fast disk copy without changing the dates in the directory. The first time this program was used many disk errors resulted. The errors were due to a consistent delay after the seek. These errors did not show up when using FLEX because the software overhead created enough delay. This was an error in our disk routines & should not cause a problem on other systems.

Sincerely,

Zeff Advanced Products

Robert Zeff
President

```
*****************************************
*        05xCLONE  VERSION 1.00         *
* THIS PROGRAM MAKES A SECTOR BY SECTOR COPY *
* OF THE DISK IN DRIVE 0 ON THE DISK IN DRIVE *
* 1. NO ERROR CHECKING IS PERFORMED AFTER THE *
* WRITE TO DISK 1.  A VALIDATE (FLEX UTILITY) *
* SHOULD BE PERFORMED ON THE NEW DISK. BY USE-*
* ING THE RAW DRIVERS & A SKEW TABLE A DISK   *
* COPY OF 2280 SECTORS CAN BE PERFORMED IN    *
* 1.5 MINUTES. THE NEW DISK MUST BE FORMATTED *
* WITH OR ANY DELETED SECTORS S MAY HAVE OLD  *
* INFORMATION ON IT. THIS PROGRAM WRITES OVER *
* EVERY SECTOR.                               *
*                                             *
*              R. ZEFF                        *
*              10-12-81                       *
*****************************************

C100                    ORG     $C100

                C003    WARMS   EQU     $C003
                C00F    OUTCH   EQU     $C00F
                CD15    GE CHR  EQU     $CD15
                CD1B    INBUFF  EQU     $CD1B
                CD24    PCRLF   EQU     $CD24
                CD33    SETEXI  EQU     $CD33
                CD39    OUTDEC  EQU     $CD39
                CD3F    RPTERR  EQU     $CD3F
                CD1E    PSTRNG  EQU     $CD1E
                DE00    READ    EQU     $DE00
                DE03    WRITE   EQU     $DE03
                0406    FMS     EQU     $D406
                DE06    VERIFY  EQU     $DE06
                DE0C    DRVSEL  EQU     $DE0C
                DE09    RESTOR  EQU     $DE09
                DE0C    SELECT  EQU     $DE0C
                C840    FCB     EQU     $C840
                0000    MEMBEG  EQU     0       buffer starts at 0.
                0004    TRACK   EQU     4       number of tracks read before a write.
                        * (This is for a 8" double sided, single density disk.)
                        * use 10 for 5 inch disk.
                004D    TRKMAX  EQU     77      (35 for 5" disk.)
                00FF    SECMAX  EQU     0FF     end of skew table marker.
C100 20  01             START   BRA     GO
C102 01                 VRM     FCB     1

C103 8E  C225           GO      LDX     #WARN
C106 BD  CD1E                   JSR     PSTRNG
C109 BD  CD15                   JSR     GETCHR  see if user wants to copy.
C10C 84  5F                     ANDA    #$5F    make it upper case
C10E 81  59                     CMPA    #'Y
```

```
C110 27  08            BEQ   OKTOGO
C112 81  4E            CMPA  #'N
C114 1027 08E8         LBEQ  WARMS
C118 20  E9            BRA   GO          incorrect response, ask again.
C11A BD  CD24  OKTOGO  JSR   PCRLF
C11D BD  CD24          JSR   PCRLF
C120 4F                CLRA              start at track 0.
C121 B7  C223          STA   TRK         start reading here.
C124 B6  C223  OVER    LDA   TRK
C127 B7  C221          STA   LSTTRK
C12A BE  C840          LDX   #FCB        select drive 0.
C12D 4F                CLRA
C12E A7  03            STA   3,X
C130 BD  DE0C          JSR   DRVSEL
C133 10BE C1FF         LDY   #DSCMAP     point to skew table.
C137 B6  05            LDA   #TRACK+1    # TRACK number of tracks.
C139 B7  C220          STA   TRKCNT
C13C 8E  0000          LDX   #MEMBEG     point to beginning of memory.
C13F E6  A0    READM   LDB   ,Y+
C141 C1  FF            CMPB  #SECMAX     see if through with this track.
C143 26  2A            BNE   NOTDON
C145 7A  C220          DEC   TRKCNT      see if ready to write.
C148 27  48            BEQ   DOWR11
C14A 7C  C223          INC   TRK         increment the track.
C14D B6  C223          LDA   TRK
C150 B1  4D            CMPA  #TRKMAX     no write if last track.
C153 27  41            BEQ   DOWR1T
C155 34  10            PSHS  X
C157 86  20            LDA   #$20        output a space.
C159 BD  CD0F          JSR   OUTCH       to keep the cursor off the number.
C15B 8E  C222          LDX   #TRK-1      point to dummy 0.
C15E 5F                CLRB
C15F BD  CD19          JSR   OUTDEC
C162 86  0D            LDA   #$D         print carriage return.
C164 BD  CD0F          JSR   OUTCH
C167 35  10            PULS  X
C169 10BE C1FF         LDY   #DSCMAP     point back to beginning sector.
C16D E6  A0            LDB   ,Y+
C16F 34  04    NOTDON  PSHS  B           save if retry is necessary.
C171 8F  C21E          STX   XTEMP
C174 B6  C223          LDA   TRK         get track.
C177 BD  DE00          JSR   READ
C17A 27  15            BEQ   NOERR
C17C 8E  C840          LDX   #FCB        point to fcb.
C17F BD  DE09          JSR   RESTOR      restore the drive.
C182 35  04            PULS  B
C184 B6  C223          LDA   TRK
C187 8E  C21E          LDX   XTEMP
C18A BD  DE00          JSR   READ
C18D 26  67            BNE   ERROR       output error message & abort.
                  err no luck on second try.
C18F 20  AE            BRA   READM
C191 35  04    NOERR   PULS  B           fix up stack.
C193 20  AA            BRA   READM
C195 86  05    DOWR1T  LDA   #TRACK+1
C197 B7  C220          STA   TRKCNT
C19A 8E  C840          LDX   #FCB
C19D 86  01            LDA   #1          select drive 1.
C19F A7  03            STA   3,X
C1A1 BD  DE0C          JSR   DRVSEL
C1A4 10BE C1FF         LDY   #DSCMAP     point to skew table.
C1A8 8E  0000          LDX   #MEMBEG     point to beginning of memory.
C1AB B6  C221          LDA   LSTTRK      point to last track read.
C1AE B7  C224          STA   WRTTRK
C1B1 E6  A0    WRITM   LDB   ,Y+
C1B3 C1  FF            CMPB  #SECMAX     see if through with this track.
C1B5 26  19            BNE   NOTDN1
C1B7 7A  C220          DEC   TRKCNT      see if through with these tracks.
C1BA 1027 FF6A         LBEQ  OVER
C1BE 7C  C224          INC   WRTTRK
C1C1 B6  C224          LDA   WRTTRK
C1C4 81  4D            CMPA  #TRKMAX
C1C6 1027 0838         LBEQ  WARMS
C1CA 10BE C1FF         LDY   #DSCMAP     point back to beginning sector.
C1CE E6  A0            LDB   ,Y+
C1D0 34  04    NOTDN1  PSHS  B           save if retry if necessary.
C1D2 8F  C21E          STX   XTEMP
C1D5 B6  C224          LDA   WRTTRK
C1D8 BD  DED3          JSR   WRITE
C1DB 27  15            BEQ   NOERR1
C1DD 8E  C840          LDX   #FCB        point to fcb.
C1E0 BD  DE09          JSR   RESTOR      restore the drive.
C1E3 35  04            PULS  B
C1E5 8E  C21E          LDX   XTEMP
C1E8 B6  C224          LDA   WRTTRK
C1EB BD  DED3          JSR   WRITE
C1EE 26  06            BNE   ERROR       output error message & abort if no
                  elues on second try.
C1F0 20  BF            BRA   WRITM
C1F2 35  04    NOERR1  PULS  B           fix up stack.
C1F4 20  B0            BRA   WRITM

C1F6 8E  C291  ERROR   LDX   #ERRMES     output an error message.
C1F9 BD  C01E          JSR   PSTRNG
C1FC 7E  CD03          JMP   WARMS

                  #skew table.
C1FF 01 06 0B 03 DSCMAP FCB  1,6,11,3,8,13,5,10,15,2,7,12,4,9,14
C20E 10 15 1A 12       FCB  16,21,26,18,23,28,20,25,30,17,22,27,19,24,29
C21D FF                FCB  $FF          marks the end.

                  #use the followins for 5" disk.
                  #DSCMAP FCB 1,3,5,7,9,2,4,6,8,10
                  # FCB $FF

C21E           XTEMP   RMB  2
C220           TRKCNT  RMB  1            track counter.
C221           LSTTRK  RMB  1
C222 00        DUMMY   FCB  0            zero mod'n for DUIDEC.
C223           TRK     RMB  1            reserve for read track counter.
C224           WRTTRK  RMB  1            reserve for write track counter.
C225 54 4B 69 53 WARN  FCC  'THIS PROGRAM MAKES AN EXACT DUPLICATE OF'
C249 0A 0D             FCB  $0A,$0D
C24F 54 4B 45 20       FCC  'THE DISK IN DRIVE 0 ON THE DISK IN DRIVE 1'
C279 0A 0A 0D          FCB  $A,$A,$D
C27C 54 59 50 45 RDY   FCC  'TYPE 'Y' WHEN READY ':4
C291 44 49 53 4B ERRMES FCC  'DISK READ ERROR, DISKCOPY ABORTED':4

                       END   GO

0 ERROR(S) DETECTED
```

Frank C. Froelich
507 So. Chestnut St.
Westfield, N.J. 07090
September 17, 1981

Dear Don,

I have been using Dynasoft Pascal for some time and feel it is an excellent compiler for the price. The use of an I/O table made it very easy to add a printer. A slight change in the supervisor added disk as well as cassete storage. The one problem I did have was in entering integers. When an integer is entered in response to a READ statement a plus sign would terminate input, also a digit that was typed wrong couldn't be corrected. Not being a good typist wrong numbers are my biggest problem. To solve this I wrote the following patch that will accept a plus before a number and recognize control H as a cancel character.

The INNUM routine is modified so that instead of getting the next character it jumps to the patch which is placed in unused memory at location zero. A plus sign will clear the sign flag and a control H will divide the number by ten, cancelling the last character entered. The backspace can be used as many times as needed, going past the first digit will simply divide zero by ten. If the fifteen bit limit is exceeded by typing the wrong character cancelling one character won't fix it. Since the sixteenth bit (the sign bit) has been changed you will have to cancel all digits and start over.

This patch works on Release 1.0 and 1.1 but can be made to work on any version by changing the INNUM locations if needed.

Sincerely,

Frank C. Froelich

```
00010                    NAM   DYNASOFT_PATCH
00020                *
00030                *This will allow Dynasoft
00040                *Pascal to recognize a
00050                *plus sign and backspace.
00060                *
00070                *The JSR INCH in INNUM is
00080                *changed to JMP PATCH which
00090                *checks for a plus or con-
00100                *trol H. If none are found
00110                *INNUM is continued. A plus
00120                *will clear the sign flag
00130                *and control H will divide
00140                *the number by ten.
00150                *
00160                      OPT   NOP,NOG
00170                *
00180 0000                 ORG   0
00190                *
00200      033F      INCH  EQU   $3F
00210      03B8      DIVIDE EQU  $3B8
00220      0039      FLAG  EQU   $39
00230                *
00240 0000 8D 3D     PATCH BSR   INCH      GET CHARACTER
00250 0002 81 2B           CMP A #'+
00260 0004 26 06           BNE   PATCH2
00270 0006 7F 0039         CLR   FLAG      POSITIVE NUMBER
00280 0009 7E 0379 PATCH3 JMP    INNUM2    RETURN TO INNUM
00290 000C 81 08   PATCH2 CMP A  #8        BACKSPACE?
00300 000E 27 03           BEQ   BACK
00310 0010 7E 037C         JMP   INNUM3    CONTINUE WITH INNUM
```

```
00320 0013 4F       BACK   CLR A
00330 0014 C6 0A           LDB B  #10      DIVIDING BY 10
00340 0016 BD 03B6         JSR    DIVIDE   CANCELS LAST DIGIT
00350 0019 20 EE           BRA    PATCH3
00360              *Changes to Interpreter.
00370 037S                 ORG    $379
00380 0379 7E 0000 INNUM2  JMP    PATCH
00390      037C    INNUM3  EQU    *        THIS IS REST OF INNUM
00400              *
00410                      END
```

DaleSoft
TOTAL E[illegible] DE60[illegible]ley Turn, Woodbridge, Va. 22193
703-670-6542

September 18, 1981

Don Williams, Editor
'68 Micro Journal
Hixson, TN

Re: DYNAMIC MEMORY AND THE GIMIX DMA DISK CONTROLLER

Dear Don,

I feel there is a need to clear up a nasty rumor for your readers.

I delayed my purchase of the GIMIX DMA DISK CONTROLLER #68 for several months because of a rumor being spread by many people, many of them usually highly knowledgeable. At least a half dozen people told me that the DMA controller would not work with my SWTPC 56K (it used to be a 16K dynamic memory board). Since I could not afford to buy the controller plus all new memroy, I was really worried and was going to opt for the lower priced (but a whole lot less versatile GIMIX programmed I/O controller.

At this point Rich Don at GIMIX, a man who really stands behind his product, came to the rescue. He didn't want me to have to do without the ability to someday upgrade to 8-inch drives, so he sold me the DMA controller under the condition that I could trade it back in for the PIO controller if it did not work with my memory. And, boy am I glad he did. I plugged it in as soon as it arrived and it hasn't dropped a bit in two months.

I am sure this success story is due to the fine engineering design provided by Rich, Bobby and Mike as well as the rest of the GIMIX staff. The phase lock data separation, write precompensation and Schmidt trigger input buffers designed to meet the data hold requirements of the Western Digital 1797 disk controller chips all pay off in excellent performance. They are well worth the price. Believe me, you'll see the difference as soon as you try one of these GIMIX controllers.

With my old controller I was always getting errors, if I was not waiting for it to make five or six attempts at reading everything. I've become extremely busy lately and I just can't afford the waiting time.

re: GIMIX FLEX, VERSION 3.4

I think as writers and editors we owe it to our readers to pass along some of the subtle improvements that are appearing in the 68XX field. Sure, the ads highlight the big points but they only touch the surface. The beginners especially will benefit if we pass along some of the points that will not fit in the ads. Take for instance, GIMIX FLEX.

You just can't believe the way the thing works. The FORMAT utility alone makes up for the fact that you have to buy a new FLEX.
Name your poison. 5-inch or 8-inch disks, single or double sided, single or double density, plus single or double stepping for the new double track drives. You can even tell it your processor speed and how many tracks you want to format. It pays to stay standard, but just for the fun of it, I formatted my old SA-400 Shugarts to 37 tracks, double density and moved from 340 to nearly 600 sectors of storage. They worked fine.

Then, there is EXTEND which lets you add space to your directory to cut down access time; CHECKSUM which lets you verify the integrity of your copies; not to mention BACKUP which copies an entire single sided 40-track disk in 40 seconds. Who said making backup copies has to be painful and take a half hour.

re: 56K ON THE SWTPC DYNAMIC MEMORY BOARD

Here's an addendum to my article in the January 1980 issue. If you want to run 56K on the 6809 and at the same time be able to run 48K on the 6800 (dropping out 8000-9FFF) do the following to to the Board Select circuit. Pick up a 74LS10 three-input NAND gate. Hook pins 1, 2 and 13 to address lines 13, 14 and 15. Hook the output of this gate (pin 12) to one input of a two input AND gate. Then, hook address line 15 and A14 NOT and A13 NOT to pins 3, 4 and 5 on the 74LS10. Hook the output of this gate (pin 6) up to one pole of a single pole, double throw switch. The center of the switch will go to the other input of the two-input AND gate. The other side of the switch will go to a 5-volt pull-up resistor. You must also put eight good 16K dynamic RAM chips in each of the four rows of sockets. Refer back to the other article for further details.

re: PULL UP RESISTORS ON THE SWTPC 6809 CPU CARD

I learned through talking to Bobby at GIMIX that it is a very good idea to put pull up resistors on the eight data lines and the four high order address lines on the SWTPC CPU card. This is especially true if you are using the card with a DMA application. It's easy to do and cheap. And, it's good insurance from garbage on the lines.

I hope some of this information will be of help to your readers. I should have the "C" review ready for you by next month.

Best Regards,

Dale L. Puckett

PS: The new Stylograph Word Processor (Version 2.0) from Bob Bundy is great.

PSS: SPELTEST (my latest effort) is the best 6809 spelling checker out. It features the good points from every review of every CP/M spelling checker on the market. Plus, it has many other unique features and leaves you with a corrected file on exit. We'll be passing along more information soon. It will be available from Frank Hogg, November 1, 1981.

Editor's Note: Reference the suggestion to pull up resistors on the SWTPC CPU Card. Having gone into this previous on some of our equipment the following is a recommended procedure.

8 high address lines 12-19 add pull up resistors
value...6.8K 1/8 watt

Read/write line add pull up also 6.8K resistor

No pull ups on data lines

I believe that the above will accomplish the DMA requirement ok and should cause no side effects in other areas of operation.

DMW - - -

U.S. POSTAL SERVICE
STATEMENT OF OWNERSHIP, MANAGEMENT AND CIRCULATION
(Required by 39 U.S.C. 3685)

1. TITLE OF PUBLICATION	A. PUBLICATION NO.	2. DATE OF FILING
68 MICRO JOURNAL	4 6 8 5 1 0	10/1/81

3. FREQUENCY OF ISSUE	A. NO. OF ISSUES PUBLISHED ANNUALLY	B. ANNUAL SUBSCRIPTION PRICE
monthly	12	18.50

4. COMPLETE MAILING ADDRESS OF KNOWN OFFICE OF PUBLICATION (Street, City, County, State and ZIP Code) (Not printers)
6131 Airways Blvd., Chattanooga, TN 37421

5. COMPLETE MAILING ADDRESS OF THE HEADQUARTERS OR GENERAL BUSINESS OFFICES OF THE PUBLISHERS (Not printers)
5900 Cassandra Smith, Hixson, TN 37343

6. FULL NAMES AND COMPLETE MAILING ADDRESS OF PUBLISHER, EDITOR, AND MANAGING EDITOR (This item MUST NOT be blank)
PUBLISHER (Name and Complete Mailing Address)
Donald M. Williams Sr., 5900 Cassandra Smith, Hixson, TN 37343
EDITOR (Name and Complete Mailing Address)
Same
MANAGING EDITOR (Name and Complete Mailing Address)
Larry E. Williams, 5900 Cassandra Smith, Hixson, TN 37343

7. OWNER (If owned by a corporation, its name and address must be stated and also immediately thereunder the names and addresses of stockholders owning or holding 1 percent or more of total amount of stock. ...)

FULL NAME	COMPLETE MAILING ADDRESS
Computer Publishing Inc.	5900 Cassandra Smith, Hixson, TN 37343
Donald M., Frances J., Larry E., Mary E., Thomas E. Williams	

8. KNOWN BONDHOLDERS, MORTGAGEES, AND OTHER SECURITY HOLDERS OWNING OR HOLDING 1 PERCENT OR MORE OF TOTAL AMOUNT OF BONDS, MORTGAGES OR OTHER SECURITIES (If there are none, so state)

FULL NAME	COMPLETE MAILING ADDRESS

9. FOR COMPLETION BY NONPROFIT ORGANIZATIONS AUTHORIZED TO MAIL AT SPECIAL RATES (Section 411.3, DMM only)
The purpose, function, and nonprofit status of this organization and the exempt status for Federal income tax purposes (Check one)
☐ (1) HAS NOT CHANGED DURING PRECEDING 12 MONTHS ☐ (2) HAS CHANGED DURING PRECEDING 12 MONTHS (If changed, publisher must submit explanation of change with this statement.)

10. EXTENT AND NATURE OF CIRCULATION	AVERAGE NO. COPIES EACH ISSUE DURING PRECEDING 12 MONTHS	ACTUAL NO. COPIES OF SINGLE ISSUE PUBLISHED NEAREST TO FILING DATE
A. TOTAL NO. COPIES (Net Press Run)	8041	8293
B. PAID CIRCULATION 1. SALES THROUGH DEALERS AND CARRIERS, STREET VENDORS AND COUNTER SALES	4158	4289
2. MAIL SUBSCRIPTION	3543	3653
C. TOTAL PAID CIRCULATION (Sum of 10B1 and 10B2)	7701	7942
D. FREE DISTRIBUTION BY MAIL, CARRIER OR OTHER MEANS SAMPLES, COMPLIMENTARY, AND OTHER FREE COPIES	129	86
E. TOTAL DISTRIBUTION (Sum of C and D)	7830	8028
F. COPIES NOT DISTRIBUTED 1. OFFICE USE, LEFT OVER, UNACCOUNTED, SPOILED AFTER PRINTING	211	265
2. RETURN FROM NEWS AGENTS	0	0
G. TOTAL (Sum of E, F1 and 2 - should equal net press run shown in A)	8041	8293

11. I certify that the statements made by me above are correct and complete — SIGNATURE AND TITLE OF EDITOR, PUBLISHER, BUSINESS MANAGER, OR OWNER: Donald M. Williams, Sr., Publisher

PS Form June 1980 3526 (Page 1) (See instruction on reverse)

Employment Opportunity

Chicago Based Terminal Systems Company has immediate opening for an experienced 6809 (Flex and OS-9) programmer with 1802 experience. Send Resume and Salary Requirements to:

CONAT INTERNATIONAL
3601 N. Runge
Franklin Park, Ill. 60131
Attn: Jeoffrey Clark
(312) 451-9191

CLASSIFIED ADS

GIMIX VIDEO BOARD 80 X 24 with RAM Character Set $300.
LEW (615) 842-4601.

***

Boards surplus after 6809 update. MP-A $50, MP-A2 $75, MP-8M (2) $80 ea. DSD 2114-16K $160, MP-B Motherboard $50.
Doug Beck (415) 948-2268.

***

SWTPC 6809, 56K, MP-S, MP-LA, $900; DC-4 ctlr, Flex9, Basic, Pascal, Forth, Sleuth, $325; SSB BFD68/2, DOS68.51C, Basic, Fortran, $850.
Charles D. Shilling (817) 461-2239.

***

TANO OutPost II, 6800 complete, 64K, two RS-232 I/O Ports, two 5" Shugart Drives, TANO Basic, COPS II (utility programs), Hemenway Editor, Compiler & Link Editor. Also Flex II ROM with TSC Basic & Basic+. Also TSC's Editor, Compiler, Debugger. Many Disks & all Manuals, $3700.
After 6:30 P.M. (205) 973-2516.

***

CASSETTE SOFTWARE: TSC 6800 Space Voyage $10, TSC Word Editor $15, AAA K Basic $15, Dynasoft Pascal $15, SWTPC 8K Basic $5, SWTPC 4K Basic $2.50. DISK: MiniFlex AAA Editor/Processor $15.
Jeffrey M. Craig (312) 225-6691.

***

HELP

Dear 68' Micro Readers,

I have gotten interested in interactive, realtime 3-D graphics. The program from SUB-LOGIC does not have the speed, source code, or resolution that I want.

Can anyone HELP me? I am interested in vector graphics packages for 6800 or 6809.

DAVID SMITH
4505 KENNEDY BLVD
N. BERGEN, N.J. 07047

I B M P A K

FLEX9™ USERS

BRIDGE the biggest GAP in your system

READ/WRITE/FORMAT IBM 3740 DISKS

No worries about ASCII -- EBCDIC conversions or different directory and file organizations
IBMPAK IS THE ANSWER!

- Lists all or part of an IBM diskette
- Copies files from IBM diskettes to FLEX diskettes
- Formats diskettes to IBM standard
- Copies files from FLEX diskettes to IBM diskettes

With IBMPAK, your FLEX9 system can access virtually all IBM and IBM compatible systems equipped with 8 inch floppy disk drives.

IBMPAK opens a new world of data for you.

CROSS THE BRIDGE
BETWEEN YOUR MICRO AND THEIR MAINFRAME

IBMPAK $125.00
(For SWTPC DMAF1 or DMAF2)
Documentation Only $15.00
Dealer Inquiries Invited

VISA/MASTER CARD PHONE ORDERS
BETWEEN 1-5 PM CST
TECHNICAL CALLS – 8-9 PM CST
512/441-6568

HELIX ENTERPRISES

504 FORT DRUM DRIVE
AUSTIN, TEXAS 78745

Terms
Check or
Money Order

*FLEX – Technical Systems Consultants, Inc.
IBM – International Business Machines
SWTPC – Southwest Technical Products Corporation

TX Residents
Add 5% tax

LUCIDATA PASCAL

If YOU need an easy-to-use, well proven implementation of Pascal, that doesn't need a mini computer to run it.... LOOK NO FURTHER - WE SELL IT!

- ONLY requires 16k-8K RAM plus one 5" disk drive
- All standard Pascal types are supported and full type-checking is performed
- REALly fast 9-digit precision scientific functions
- Optional allocation of variables to absolute memory locations allows easy control of memory-mapped peripherals (eg video boards, PIA's ACIA's etc)
- Fully optimised run-time systems for 6800 and 6809
- Any number of EXTERNAL user supplied routines may be easily linked to Pascal defined function and procedure identifiers within your program
- Fast, easy-to-use compiler generates ultra efficient position independent and ROMable P-Code instructions
- Multiple files - may be sequential/random disk files supported by your DOS, or physical devices added by you
- Scalar I/O to READ and WRITE your own SCALAR TYPES
- Custom versions available to special requirements
- All LUCIDATA Pascals are supplied with a comprehensive User Manual PLUS lots of demo programs and are fully supported by our Update Service

6800 FLEX 2 version on 5" disk $150, on 8" disk $165
6809 FLEX 9 version on 5" disk $190, on 8" disk $205
User Manual separately $25 (deductable from purchase)
Prices include Airmail postage ANYWHERE in the WORLD!
Payment by Mastercard/VISA or certified cheque
Despatched by return of post (UK → East Coast 5 days)

Lucidata

LUCIDATA Ltd., P.O. Box 128
Cambridge, CB2 5EZ, ENGLAND
Telephone Orders (0223) 841906

Purveyors of Pascal since 1979

LUCIDATA Pascal is also available for Heath HDOS from POLYBYTES, 325 19th Street SE., Cedar Rapids, IA 52403 & Smoke Signal Broadcasting DOS68D from WINDRUSH Micro Designs, Gaymers Way, North Walsham, NR28 0AN, ENGLAND

STYLOGRAPH™

6809
WORD PROCESSING SYSTEM

STYLOGRAPH 2.0

All of the convenience and features for which Stylograph is well known plus:

- True proportional printing on specialty printers.
- Files longer than memory.
- "Help" command to aid in learning.
- New menu driven, self prompting functions.
- Left and right scrolling for pages larger than screen.
- Embedded printer control commands allowed.
- Simplified method for underline, boldface, superscript, etc.
- Supports NEC, Diablo, Qume, 737, and 739 printers.
- Can be user configured for virtually any terminal or printer.

$295, manual $15, updates from old versions $180.

STYLOGRAPH 3.0

This version is designed for "tty" printers but is otherwise identical to version two. It does not support specialty characteristics such as superscript, subscript, incremental printing, and proportional printing.

$195, manual $15, updates from old versions $90.

STYLOGRAPH MAIL MERGE

This program takes files of variables, such as names and addresses, and inserts them into a Stylograph text file for automated mail list generation. It will also allow a number of Stylograph text files to be appended at printout time so that page numbers and headings will be continuous in the printout.

$125, manual $10.

STYLOGRAPH SPELLING CHECKER

This is a valuable addition to any word processing application. It checks all words in a manuscript against an internal dictionary. The dictionary included has a vocabulary of over 20,000 words and is fully expandable. New words encountered in the text may be added to the dictionary making the creation of custom tailored and foreign language dictionaries a snap.

$145, manual $10.

When ordering specify operating system (FLEX™, Uniflex™) and disk size. VISA & MC accepted.

20% discount on 3 program order.

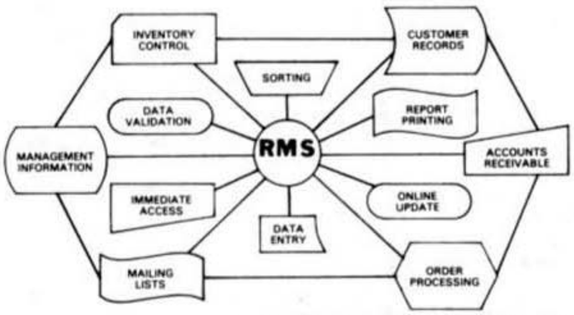
STYLO SYSTEMS

BOX 238 WILLIAMSVILLE, NY 14221
716—634-2466

6809 RECORD MANAGEMENT SYSTEM RMS

DATABASE MANAGEMENT

- USER DEFINED RECORD FORMAT VIA DATA DICTIONARY
- SCREEN ORIENTED, FORM FILL OUT TYPE OF ACCESS
- OPTIONAL TWO LEVEL RECORD HIERARCHY
- ALL FILES IN ASCII TEXT FORMAT, BASIC COMPATIBLE
- DIRECT ACCESS BY KEY FIELD, MULTIPLE INDEX FILES
- EXTENSIVE DOCUMENTATION, SAMPLE APPLICATION
- VERSATILE, PROFESSIONAL QUALITY REPORT WRITER
- BUILT-IN SORT/MERGE
- EASY TO USE



RMS is a complete DATABASE MANAGEMENT package for the 6809 computer. It is made up of five machine language programs that make up the most powerful business programming tool available for the 6809. It can be used by the relative novice, to implement an incredible variety of information storage and retrieval applications, without any programming. However, the programmer can use RMS as part of the solution to a larger problem, saving many hours of unnecessary program development time. RMS can be used to handle data input, editing, validation, on-line retrieval, sorting and printed reports. Custom data manipulation can be filled in by the user's BASIC programs.

SINGLE CPU LICENSE

FLEX*	$200
OS-9+	$250
UNIFLEX*	$300

TERMS: VISA/MC/PREPAID

WASHINGTON COMPUTER SERVICES
3028 SILVERN LANE
BELLINGHAM, WA 98225
1 (206) 734-8248

* FLEX and UNIFLEX are trademarks of Technical System Consultants Inc.; + OS-9 is a trademark of Microware

ACTION GAMES

The fastest growing producer of computer games for your 68XX has the products you have waited for!!

NEW! COLOR BERSERK

AN ARCADE GAME FOR THE COLOR COMPUTER

Fast paced action • Super Hi-Res Graphics
Dynamite sound effects • One or two players • Runs in 16K of memory
This game will astonish you with its Detail and Quality
It sets a standard for others to follow

— ADVENTURES —

Calixto Island • The Black Sanctum

Highly acclaimed by reviewers • Challenging situations
Fast, efficient machine language • Runs in 16K of memory
Save game in progress

Adventures on 5¼ TSC FLEX disc (specify 6800 or 6809)	ea. $24.95
Both adventures on single disc	$39.95
Adventures for color computer	ea. $19.95
Color Berserk for color computer	ea. $24.95

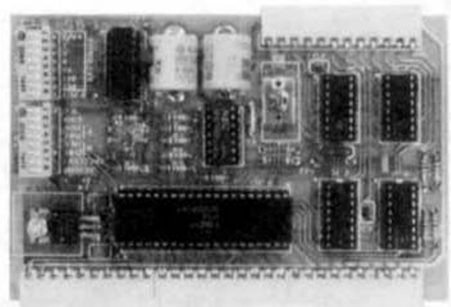
Shipped prepaid in continental U.S. California residents, please add 6% tax

VISA — MORE COMING SOON — master charge

MARK DATA PRODUCTS

(714) 768-1551 Evenings

CALENDAR-CLOCK / TIMER / PARALLEL PORT



Calendar - Clock CLK68-1

Interval Timer

Parallel I/O Port

Construction

Manual -- Well documented - 36 pages

Dealer & OEM discount available

Assembled and tested	$119.95	Kit	$89.95
Goldplated bus conn	7.50	2 MHz option	2.50
Disk 5 or 8 in. SSD of Flex* OS-9 Available NOW			14.95

ROBERTSON ELECTRONICS
1003 Warm Sands Dr. SE
Albuquerque, NM 87123

Phone (505) 294-0025
NM residents add 4% tax
Add $3 Shipping & Handling

6809 Color Computer

EDITOR, ASSEMBLER AND MORE!

TIRED OF WAITING FOR SOFTWARE DELIVERY?

WE SHIP FROM STOCK!

NEW!

SOFTWARE DEVELOPMENT SYSTEM

The Micro Works Software Development System (SDS80C) is a complete 6809 editor, assembler and monitor package contained in one Color Computer program pack! Vastly superior to RAM-based assemblers/editors, the SDS80C is non-volatile, meaning that if your application program bombs, it can't destroy your editor/assembler. Plus it leaves almost all of 16K or 32K RAM free for your program. Since all three programs, editor, assembler and monitor are co-resident, we eliminate tedious program loading when going back and forth from editing to assembly and debugging!

The powerful screen-oriented Editor features finds, changes, moves, copys and much more. All keys have convenient auto repeat (typamatic), and since no line numbers are required, the full width of the screen may be used to generate well commented code.

The Assembler features *all* of the following: complete 6809 instruction set; complete 6800 set supported for cross-assembly; conditional assembly; local labels; assembly to cassette tape or to memory; listing to screen or printer; and mnemonic error codes instead of numbers.

The versatile ABUG monitor is a compact version of CBUG, tailored for debugging programs generated by the Assembler and Editor. It features examine/change of memory or registers, cassette load and save, breakpoints and more.

SDS80C Price: $89.95

CRACK THOSE ROMS!

SOURCE GENERATOR: This package is a disassembler which runs on the color computer and enables you to generate your own source listing of the BASIC interpreter ROM. Also included is a documentation package which gives useful ROM entry points, complete memory map, I/O hardware details and more. Disassembler features include cross-referencing of variables and labels; output code which can be reassembled; output to an 80-column printer, small printer or screen; and a data table area specification which defaults to the table boundaries in the interpreter ROM. A 16K system is required for the use of this cassette.

80C Disassembler Price: $49.95

LEARN 6809!

6809 Assembly Language Programming, by Lance Leventhal, contains the most comprehensive reference material available for programming your Color Computer.

Price: $18.95

PARALLEL O!

USE A PARALLEL PRINTER with your Color Computer! Adaptor box plugs into the serial port and allows use of Centronics/Radio Shack compatible printers with parallel interface. Assembled and tested.

P180C Price: $69.95

CBUG IS HERE!

MONITOR TAPE: A cassette tape which allows you to:

- Examine or change memory using a formatted hex display
- Save areas of memory to cassette in binary (a "CSAVEM")
- Download/upload data or programs to a host system
- Move the video display page throughout RAM
- Send or receive RS-232 at up to 9600 baud
- Investigate and activate features of your computer, such as hi-res graphics or machine language music
- Use your color computer as an intelligent peripheral for another computer, a color display or a 6809 program development tool

The monitor has 19 commands in all, and is relocatable and re-entrant.

CBUG Tape Price: $29.95

MONITOR ROM: The same program as above, supplied in 2716 EPROM. This allows you to use the entire RAM space. And you don't need to re-load the monitor each time you use it. The EPROM plugs into the Extended Basic ROM Socket or a modified ROMPACK.

CBUG ROM Price: $39.95

32K RAM!

MEMORY UPGRADE KITS: Consisting of 4116 200ns. integrated circuits, with instructions for installation. 4K-16K Kit Price: $39.95. 16K-32K Kit (requires soldering experience) Price: $39.95

THE MICRO WORKS

MasterCharge/Visa Accepted

California residents add 6% tax.

P.O. BOX 1110, DEL MAR, CA 92014 (714) 942-2400

ALFORD AND ASSOCIATES - GOOD NEWS!

COLOR COMPUTER TALKS!

Alford & Associates is now shipping its SP-1 "SPEAKER PACK" for the Radio Shack Color Computer. The SP-1 plugs directly into the ROM-Pack slot, and its use requires no wiring, computer modification or electronics knowledge!

The software provided gives your computer the power of speech using nothing more than Basic Peek and Poke statements! The SP-1 can add a new dimension to your games, business programs or CAI drills. Just about any application can benefit from the SP-1!

The SP-1 allows unlimited speech. Also, the SP-1 requires less memory overhead for speech than any other type of unlimited voice synthesizer on the market today. Typically, fewer bytes of storage are needed than the equivalent number of letters in English Text! Basic data statements suffice to store most any text you want! Also, no machine language routines are needed to drive it!

The SP-1 comes with sample software in Basic to demonstrate the power of this fantastic device. Alford's even supplies a version of their VOX-EDITOR to allow users who have 16K of memory to edit speech files quickly and easily (Extended Basic is NOT required)!

The SP-1 includes a comprehensive manual which provides speech theory, use of the included software, phoneme code charts, sample programs and much more!

COMPUTERS SPEAK TEXT!

Alford's has been on the lookout for a good speech convertor program for use with their SP-1 and VS-1 speech synthesizers, and report having finally found it! The program takes English text in ASCII form, converts it, and then directly drives the synthesizer! By sending the appropriate control codes, you can even switch between text conversion or speaking letters or speaking straight phoneme information!

The standard version is designed to work just like OUTCH or OUTEE. In all modes, you call the converter with characters in the A-REGISTER. In conversion mode, characters are accumulated until a word is complete, then it is converted and spoken. In letter mode each character is treated as an ASCII character and pronounced. In phoneme mode, each character is treated as a single inflected phoneme code.

The Color Computer version is even easier to use. Once loaded, Basic has a new verb -- SAY. To use the converter, you simply use the command to make the computer talk; i.e., SAY "THIS IS A COMPUTER SPEECH TEST", and the Color Computer will!

TREK-69, COLOR-TREK

Ever since Alford's introduced TREK-68 last year, we've been asking them to do a version for 6809 users and for the new TRS-80 Color Computer. They said that they could have, but their old home-brew computer and a bright shiny store-bought one might not have been compatible. Rather than take a chance, they held off until they decided on which systems they wanted to buy. Well, they finally decided, and at last, our wishes have come true!

If you have a MEMORY-MAPPED DISPLAY then TREK-68 or TREK-69 are for you. If you have a color computer with 16K memory (Extended Basic not required), then you should ask for COLOR-TREK.

They took the classic trek-game and rewrote it completely in assembly code, making it run in REAL-TIME. The result is, we feel, the finest TREK game available!

Enemy destroyers chase you even as you move about the quadrant. Their multiple battle plans make them hard to evade! You dodge torpedoes and return fire. Damage is sustained and repairs occur as you play. Messages flash on and off. Again, all in REAL TIME!

Game difficulty levels run from SIMPLE (for beginners) to a level which, to our knowlege, only one person other than the author himself has succeeded in winning! This is not a simple game. The best winning time recorded for the awful SUICIDE OPTION is twenty-nine minutes. The fastest loss we have seen occurred in only eleven seconds!

Have a MEMORY-MAPPED DISPLAY? Then all we can ask is, why haven't you TREK-ed?!

SPEAKER BARE BOARDS!

Alford and Associates recently lowered the price on its VS-1 synthesizer. A company spokesman stated that the reduction was due to the great response that the SS-50 community has given the board. Now they are going one step further. You can now buy a bare board, manual, disk or synthesizer chip seperately. See the price list for details.

Alford & Assoc.
P.O. Box 6743
Richmond, Va.
23230
804-320-6722

NEW SCREEN EDITOR!!

Does your terminal have an addressable cursor like the Soroc IQ-120? Does your terminal scroll when you do a new line on the bottom line? Does your terminal run full-duplex? If not, then you should skip this ad. Otherwise, you may be ready for SCREDITOR III!

How would you like an editor that will handle a 252-column spread sheet? Or one that allows you to move margins anywhere and at any time? Or that handles true multi-column edit jobs like this page? Or that formats text as you type? Or that allows you to set or clear tabs at any time, anywhere, with a single keystroke? SCREDITOR III CAN!

Like to be able to define what single-key operations you do with what single keys? Or for that matter, what command names you want to give the commands? YOU CAN WITH SCREDITOR III!

Wouldn't it be nice to be able to define up to twenty-six editing macros, with a macro length of up to 1000 characters? Or to be able to mix commands, operations and text, all in the same macro? Or even to display and edit the macros themselves just like text? Or save and load your macros from disk files? YOU CAN WITH SCREDITOR III!

How about file handling. Would you like to edit unlimited-sized files? Or to be able to read selected lines out of one file into another? Or how about conditional previewed reading to let you see the lines before inserting them? Or be able to write lines out to new files? Or to specify where to start reading or writing, and how much at a time, and how many times? YOU CAN WITH SCREDITOR III!

Think about it. Thirty-two control-code operations. About fifty other commands, and the number is growing. Twelve justification commands alone! ONLY WITH THE ALL-NEW SCREDITOR III.

As if all of this, and much more than we have room for here, isn't enough, this new editor is available for FLEX 1.0, FLEX 2.0, FLEX-9, DOS68 and DOS69! OS-9 versions to be available soon (maybe as you read this, even!).

In talking to John Alford, proprietor of Alford and Associates, we were told that he is tired of writing editors. He indicates that he knows only two ways to stop; get out of the business, or write the ultimate editor. It doesn't appear that he is going out of business soon!

If this hasn't convinced you that you should be using SCREDITOR III, then call or write for more details, or for the complete SCREDITOR III spec sheet. Our only question is, why continue to edit, when you can SCREDIT?

SCREDITOR III is available for most serial terminals, and all memory-mapped displays. 6800 and 6809 versions are ready now!

SSB DOS UTILITIES

As many of you know, Smoke Signal Broadcasting's DOS is one of the best around. There was, however, one thing we felt to be lacking...disk names!

Alford and Associates has finally found how to go about naming a disk in a non-destructive and secure way, and started to write some utilities using the disk information record. Their NAME utility allows you to name your disks. The information sector includes the disk name, serial number, creation date, last update date, a comment field, and last but not least, a disk file access code.

The access code led them to the second program, LOCK. With this program you can write, delete, and LIST LOCK your files!

With their LIST program, the list-locked files do not list unless you give the access code for the disk! In addition, you do not have to look at a pile of transient commands unless you want to, as LIST allows you to option the listing for certain files. LIST even lets you list the disk information record!

The UPDATE program lets you change the information record! They also include PURGE to clean up disks, DUMP to make pretty core images, and TITLE to print title pages on all of your listings.

The UTILITIES #1 disk is available for DOS68, versions 4.0 and up, and for all versions of DOS69. The manual itself is a good reason for buying this package, as it has a batch of information on SSB disk structure.

PRICE LIST

SP-1 SPEAKER-PACK, COMPLETE . .	$179.95
SP-1 MANUAL ONLY	19.95
VS-1 SPEAKER, COMPLETE	$189.95
VS-1 BARE BOARD ONLY	19.95
VS-1 MANUAL ONLY	19.95
VS-1 VOX-EDIT PROGRAM DISK ONLY	24.95
VS-1 MANUAL, BOARD, CHIP, DISK	109.95
VS-1 ASSEMBLED, LESS SC-01 . .	129.95
SC-01A SYNTHESIZER CHIP	65.00
SC-1/6800 SPEECH TRANSLATOR . .	$29.95
SC-1/6809 SPEECH TRANSLATOR . .	29.95
SC-1/COLOR SPEECH TRANSLATOR .	19.95
TREK-68 DISK, MANUAL	$24.95
TREK-69 DISK, MANUAL	24.95
COLOR-TREK CASSETTE, MANUAL . .	14.95
MANUAL ONLY, ANY OF ABOVE . . .	9.95
SSB DOS UTILITIES #1 DISK . . .	$24.95
SCREDITOR III/6809 DISK, MANUAL	$89.95
SCREDITOR III/6800 DISK, MANUAL	89.95
SCREDITOR III MANUAL ONLY . . .	24.95

PRICE OF MANUAL REFUNDABLE ON ANY ORDER. PRICES SUBJECT TO CHANGE WITHOUT NOTICE. BE SURE TO SPECIFY DISK SIZE, OPERATING SYSTEM AND PROCESSOR TYPE WHEN ORDERING.

THE LAST WORD...

This year (our third) has been the most successful ever. I would like to take an opportunity to thank our many fine customers for their support and understanding.

Most of all though, I would like to express my thanks to God, who is the major "Associate" in my business. At this time of year it seems especially appropriate to pause to give thanks and to remember the many miracles which He has performed for all of us. Too often, we take His grace for granted.

In this season, we celebrate the miracle of the eternal light and the miracle of the birth of the Messiah. Both stand for hope for the human race. Both stand for God's light in our life. And both show His love for us, in that He provides for our needs in many ways, especially in times of darkness.

I pray that in this next year, my walk with Him will lead me closer to the place He wants me, and that I might become even better able to serve every one of you, my customers. I also hope that each one of you will pray for me and my family so that we might grow as He wants us to, and that we might be more worthy of your patronage.

Thank you again, in love and in prayer, from all of us, and especially from....

John L. Alford (proprietor)
Sally Anne Alford (most everything else)
Alford and Associates

GENERAL INFO

All of Alford's software is available on 5- or 8-inch disk except where noted. Also, except where noted, all software is available for FLEX 1.0, FLEX 2.0, FLEX-9, DOS68 or DOS69. Versions for OS-9 are coming soon. Software orders are normally shipped within three days. Hardware runs from stock to 30 days.

You should add $5 for shipping on any order under $100. Alford's pays shipping over $100. Overseas orders, add $10 for air mail delivery. Virginia residents add 4% sales tax. Any order received without shipping or tax (where applicable) will be returned unfilled. Unless you specify otherwise, shipping is by UPS in the U.S.

Alford's accepts Mastercard, Visa, COD's or checks in U.S. funds. Open account orders by prior arrangement only. Personal checks may delay shipping by two or three weeks.

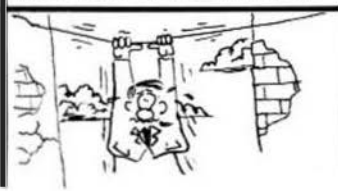
Flex is a trademark of Technical Systems Consultants. TRS-80 and Color Computer are trademarks of the Tandy Corporation. OS-9 is a trademark of Microware Systems Corporation.

Most people don't realize it, but the total national debt in this country is just now passing one trillion dollars. Today's economists tell us that this is why we have inflation. What they don't tell you is that the total debt in the private sector is over nine trillion dollars! Th[illegible]nterest on the private [illegible]nts to appr[illegible]mately [illegible]al gr[illegible]nal

RICHMOND (NP). Today, 46 year old Harley R. Kumquat was observed biting a mongrel dog in Monroe Park. When asked by this reporter what prompted such a singular activity, Harley replied that he had been coming to the park dailey for fifteen years, and that on each and every outing, the dog had b[illegible]tten him. At l[illegible]e could stand no [illegible]d b[illegible] in kind.

FOGGY BOTTOM, AR -- This reporter was previously of the opinion that he had seen everything, but found that there is truly something new under the sun, as the honorable Senator Bilge Pump was caught in the very act of telling the truth to his constituants.

When queried about his uncommon fau[illegible] the senator [illegible]d that he d[illegible] Fur



Software piracy costs each of us. Most people don't realize that the reason that much of the available software costs what it does simply because the writer has to increase the price of his product to make up for the lost sales which result from people who think they are doing a goo[illegible]ed for a friend. A [illegible]rvativ[illegible] is that [illegible]

SMOKE SIGNAL BROADCASTING

31336 VIA COLINAS WESTLAKE VILLAGE, CA 91362 TEL (213) 889-9340 / TWX 910-494-4965

MICROPRODUCT BULLETIN

ARE YOU REALLY SATISFIED?

. . . with the **PRODUCTS** you are selling?
. . . with the **PROFITS** you are making?
. . . with the **SUPPORT** you receive?

Let's face it—

The PRODUCTS, the PROFITS, and the SUPPORT are the key factors in a Dealer-Manufacturer relationship.

IF YOU WANT INCREASED SALES, BIGGER PROFITS, AND REAL SUPPORT — CALL SMOKE SIGNAL BROADCASTING.

We know you've heard that story before. In fact, it probably compelled you to become a dealer in the first place! What you probably **didn't** know is this:

Too many computer companies make ambitious claims when you first express interest, but **very few of them will continue to hold up their end of the deal once you have signed up**.

Smoke Signal **will** deliver what is promised . . . plus much more!

Being a Smoke Signal Broadcasting Dealer is easy, we **don't** require large initial orders; we **don't** impose monthly quotas or contracts on you. What we **do** have are good, reliable, high-quality computer systems; excellent pricing and large discounts. Those factors alone provide enough dealer incentive **without** messy monthly quotas.

Our business computers, the CHIEFTAIN Series, are state-of-the-art 6800/6809 and SS-50 Bus. We run FLEX, OS-9 Multi-user, plus our own DOS68D and DOS69D, with absolutely no changes on your part. Our marketing plans include you, the Dealer — our **real** sales force.

Convince yourself. Call or write Deborah Conrad (213) 889-9340 for more Dealer and OEM information.

6809 Relocating Recursive Macro Assembler & Loader/Linker with text editor

- Runs on 6809 system (in as little as 32k)
- Interactive or non-interactive (batch) mode
- Supports relocatable and absolute code
- Can assembler 6800 & 6801 source and generate 6809 object
- CROSS ASSEMBLER MODE—can assemble 6800 & 6801 source and generate 6800 & 6801 object
- 8 character global and local labels
- 76 English error messages
- Alphabetized or non-alphabetized symbol table
- Cross reference table
- Checks for unreferenced labels
- Can enable the insertion of a 'SWI' after every instruction
- Supports the following assembler directives: ASECT, CSECT, PSECT, DEBUGS, END EQU, EXEC, EXT, FCB, FCC, ID, INT INCLUDE, NAM, OPT, ORG, PAG, RMB SET, SETDP, SPC, TTL, FDB
- Program sectioning (ASECT, CSECT, or PSECT)
- Source of I/O routines supplied
- Total FLEX* compatability
- Powerful co-resident text editor
- TSC source file compatibility

Powerful Macro Capabilities

- 8 character symbolic (substitution) labels
- branch on conditions EQ, NE, LE, LT GT, GE
- supports nesting
- Logical, aritmetic, and string labels
- Sublist processing
- Powerful string functions such as SUBSTRING
- Supports the following directives ACNT, AGO, AIF, ANOP, GBLA, GBLB, GBLC LCLA, LCLB, LCLC, MACRO, MEND, MNOTE SETA, SETB, SETC

Loader/Linker

- Link and load (with offset)
- Link and save
- Global cross references
- Supports the following commands LOAD, COMPARE, SAVE, EDIT, LINK, MAP INTERNALS, UNDEFINED, CLEAR, RUN SECTION, FIX

MASM 6809 ---------- **$ 250.00**

User's Manual Only (about 200 pages—refundable)— **$ 25.00**

A version of the above assembler which generates ABSOLUTE code is also available

ASMB 6809 ---------- **$ 150.00**

The above software is available on 5 or 8 inch FLEX* disks, prices include one year maintenance (single CPU). Even if you already own an assembler you should seriously consider ordering these powerful tools.

CINCITEK Software

Box 19365
Cincinnati, Ohio 45219

(513) 751-6203 (evenings)

Ohio residents add 4.5 % sales tax
Add 2 % for postage
Foreign orders add 10 %

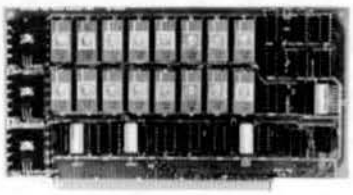
Dealer Inquires Welcome

*FLEX is a trademark of TECHNICAL SYSTEM CONSULTANTS

DIGITAL RESEARCH COMPUTERS
(214) 271-3538

ALL SALES ARE MADE SUBJECT TO THE TERMS OF OUR 90 DAY LIMITED WARRANTY. A COPY OF THIS WARRANTY IS AVAILABLE FREE, ON REQUEST.

32K S-100 EPROM CARD
NEW!



$79.95 KIT

USES 2716's

Blank PC Board - $34

ASSEMBLED & TESTED ADD $30

SPECIAL: 2716 EPROM's (450 NS) Are $9.95 Ea. With Above Kit.

KIT FEATURES:
1. Uses +5V only 2716 (2Kx8) EPROM's.
2. Allows up to 32K of software on line!
3. IEEE S-100 Compatible.
4. Addressable as two independent 16K blocks.
5. Cromemco extended or Northstar bank select.
6. On board wait state circuitry if needed.
7. Any or all EPROM locations can be disabled.
8. Double sided PC board, solder-masked, silk-screened.
9. Gold plated contact fingers.
10. Unselected EPROM's automatically powered down for low power.
11. Fully buffered and bypassed.
12. Easy and quick to assemble.

32K SS-50 RAM

$299.00 KIT

For 2MHZ Add $10

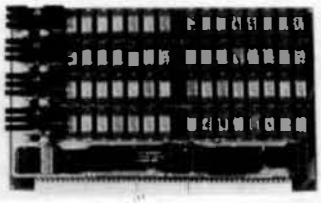
Blank PC Board $50

For SWTPC 6800 - 6809 Buss

Support IC's and Caps $19.95

Complete Socket Set $21.00

Fully Assembled, Tested, Burned In Add $30



At Last! An affordable 32K Static RAM with full 6809 Capability.

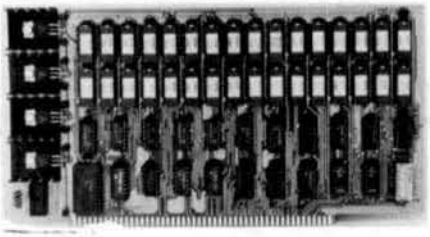
FEATURES:
1. Uses proven low power 2114 Static RAMS.
2. Supports SS50C - EXTENDED ADDRESSING.
3. All parts and sockets included.
4. Dip Switch address select as a 32K block.
5. Extended addressing can be disabled.
6. Works with all existing 6800 SS50 systems.
7. Fully bypassed. PC Board is double sided, plated thru, with silk screen.

16K STATIC RAM KIT-S 100 BUSS

PRICE CUT!

$169.95 KIT

FOR 4MHZ ADD $10



KIT FEATURES
1. Addressable as four separate 4K Blocks.
2. ON BOARD BANK SELECT circuitry. (Cromemco Standard!). Allows up to 512K on line!
3. Uses 2114 (450NS) 4K Static Rams.
4. ON BOARD SELECTABLE WAIT STATES.
5. Double sided PC Board, with solder mask and silk screened layout. Gold plated contact fingers.
6. All address and data lines fully buffered.
7. Kit includes ALL parts and sockets.
8. PHANTOM is jumpered to PIN 67.
9. LOW POWER under 1.5 amps TYPICAL from the +8 Volt Buss.
10. Blank PC Board can be populated as any multiple of 4K.

BLANK PC BOARD W/DATA-$33
LOW PROFILE SOCKET SET-$12
SUPPORT IC'S & CAPS-$19.95
ASSEMBLED & TESTED-ADD $35

OUR #1 SELLING RAM BOARD!

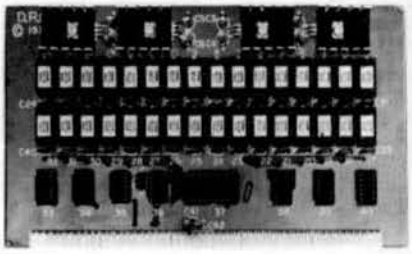
16K STATIC RAM SS-50 BUSS

PRICE CUT!

$159 KIT

FULLY STATIC!

FOR 2MHZ ADD $10



FOR SWTPC 6800 BUSS!

ASSEMBLED AND TESTED - $35

KIT FEATURES
1. Addressable on 16K Boundaries
2. Uses 2114 Static Ram
3. Fully Bypassed
4. Double sided PC Board. Solder mask and silk screened layout
5. All Parts and Sockets included
6. Low Power. Under 1.5 Amps Typical

BLANK PC BOARD—$35 COMPLETE SOCKET SET—$12
SUPPORT IC'S AND CAPS—$19.95

NEW! **STEREO! S-100 SOUND COMPUTER BOARD** NEW!

At last, an S-100 Board that unleashes the full power of two unbelievable General Instruments AY3-8910 NMOS computer sound IC's. Allows you under total computer control to generate an infinite number of special sound effects for games of any other program. Sounds can be called in BASIC, ASSEMBLY LANGUAGE, etc.

COMPLETE KIT! $84.95 (WITH DATA MANUAL)

BLANK PC BOARD W/DATA $31

KIT FEATURES:
- TWO GI SOUND COMPUTER IC'S.
- FOUR PARALLEL I/O PORTS ON BOARD.
- USES ON BOARD AUDIO AMPS OR YOUR STEREO.
- ON BOARD PROTO TYPING AREA.
- ALL SOCKETS, PARTS AND HARDWARE ARE INCLUDED.
- PC BOARD IS SOLDERMASKED, SILK SCREENED, WITH GOLD CONTACTS.
- EASY, QUICK, AND FUN TO BUILD, WITH FULL INSTRUCTIONS.
- USES PROGRAMMED I/O FOR MAXIMUM SYSTEM FLEXIBILITY.

Both Basic and Assembly Language Programming examples are included.

SOFTWARE:

SCL™ is now available! Our Sound Command Language makes writing Sound Effects programs a SNAP! SCL™ also includes routines for Register-Examine-Modify, Memory-Examine-Modify, and Play-Memory. SCL™ is available on CP/M® compatible diskette or 2708 or 2716 Diskette - $24.95. 2708 - $19.95. 2716 - $29.95. Diskette includes the source. EPROM'S are ORG at E000H. **(Diskette is 8 inch Soft Sectored)**

SPECIAL PURCHASE!

UART SALE!

TR1602B — SAME AS TMS6011, AY5-1013, ETC. 40 PIN DIP

TR1602B

$2.95 EACH 4 for $10.00

CRT CONTROLLER CHIP

SMC #CRT 5037. PROGRAMMABLE FOR 80 x 24, ETC. VERY RARE SURPLUS FIND. WITH PIN OUT. $12.95 EACH.

4K STATIC RAM

National Semi. MM5257. Arranged 4K x 1. +5V, 18 PIN DIP. A Lower Power, Plug in Replacement for TMS 4044. 450 NS. Several Boards on the Market Will Accept These Rams. SUPER SURPLUS PURCHASE! PRIME NEW UNITS!

8 FOR $16 32 FOR $59.95

***NEW!* G.I. COMPUTER SOUND CHIP**

AY3-8910. As featured in July, 1979 BYTE! A fantastically powerful Sound & Music Generator. Perfect for use with any 8 Bit Microprocessor. Contains: 3 Tone Channels, Noise Generator, 3 Channels of Amplitude Control, 16 bit Envelope Period Control, 2-8 Bit Parallel I/O, 3 D to A Converters, plus much more! All in one 40 Pin DIP. Super easy interface to the S-100 or other busses. **$11.95 PRICE CUT!**

SPECIAL OFFER: ~~$14.95~~ each Add $3 for 60 page Data Manual

Digital Research Computers (OF TEXAS)

P.O. BOX 401565 • GARLAND, TEXAS 75040 • (214) 271-3538

TERMS: Add $2.00 postage. We pay balance. Orders under $15 add 75¢ handling. No C.O.D. We accept Visa and MasterCharge. Tex. Res. add 5% Tax. Foreign orders (except Canada) add 20% P & H. Orders over $50, add 85¢ for Insurance.

*TRADEMARK OF DIGITAL RESEARCH.

WE ARE NOT ASSOCIATED WITH DIGITAL RESEARCH OF CALIFORNIA, THE SUPPLIERS OF CPM SOFTWARE.

SOFTWARE SYSTEMS AND SOLUTIONS FROM SD

Software Dynamics has been offering quality software for 68xx class machines since 1977. We recognize that our customer's needs grow, and design our software to provide a growth path. SD offers compatible single-user, multi-user, and (soon) network systems to meet this need. Programs that run on the single-user system run unmodified on the multi-user and network systems. A virtual terminal driver assures that screen-oriented applications run on any CRT.

The BASIC compiler is the result of 10 years experience in building BASIC interpreters and compilers. Sequential, random and indexed files are supported, along with true BCD numbers to eliminate conversion errors. Long variable names, blocks for structured programming, and true multi-parameter subroutines and string functions aid program construction and maintenance. Compiled programs are both small and very fast, and the source is completely protected.

SD also offers word processing and accounting software built to the same high standards as its system software.

SDOS™

- Interrupt-driven DOS
- Read-ahead on sequential files
- LRU buffering optimizes random files
- Byte-addressable, device independent files
- Virtual terminal driver handles any CRT
- Command files
- Keyboard typeahead
- Disk file structure validation program
- Adaptable to any 68xx micro with 40kb or more
- Any combination of floppies or hard disk

SDOS/MT™

- Multi-user version of SDOS
- 1 to 8 users
- User space to 60 kilobytes
- Any hardware mapping technology

Structured Design BASIC V1.4

- 32 character variable names
- Line labels
- Parameterized, multi-line functions and subroutines
- Full access to SDOS sequential and random file facilities
- Multi-key indexed file option
- Print using with floating dollar sign
- 10 digit fast decimal floating point (no conversion errors!)
- Binary integer arithmetic
- IF-THEN-ELSE, WHILE-DO, ON ERROR DO
- Many other block structure facilities
- COMMON and program chaining
- Super fast execution
- Very compact compiled code
- Complete error trapping

SDOS is available for the following hardware:

Midwest Scientific Instruments 6800
Pace Technology 480
WaveMate Series 2000
Omnibyte 800, 890
Brittania Computing Models 242, 363, 484, 1010
Motorola Exorcisor™

SEDIT

- What-you-see-is-what-you-get editor
- Uses cursor and arrow keys for positioning
- Insert by typing at cursor location
- Delete by RUBOUT at cursor location
- Edit any size file
- Cut and paste to move text
- Automatic margin wrap
- Tabs
- Very easy to learn

TYPE

- Word or document processing
- Letter and envelope generation
- Form letters with mailing lists
- Complete margin justification
- Centering and underscoring
- Table of contents generation

COUNT/UP

- Full accounting package
- General Ledger, Accounts Receivable, Accounts Payable
- User specifies all report formats
- Account structure and formats changeable at any time

Other programs:

SORT	Sorts records according to any combination of key fields
EDIT	Powerful context editor
IDB	ROMable debugger with single step capabilities
MEMTEST	Thorough memory diagnostic
ASM	6800/6809 assemblers
CHESS	For lighter moments

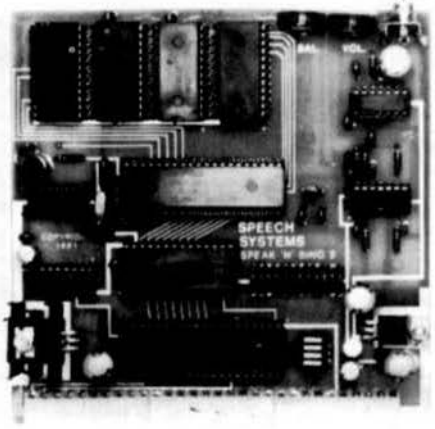
BASIC 1.3, ASM and EDIT are available for 6800 FLEX™, SSB DOS or MDOS™.
SD also offers both 5¼" and 8" Winchester disk drives for Exorcisor systems.

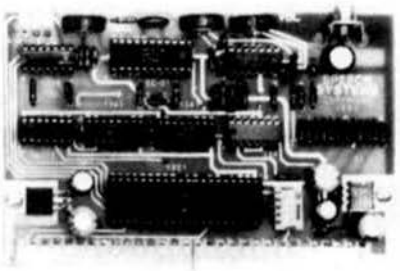
Write for free catalog.

SOFTWARE DYNAMICS
2111 W. Crescent, Suite G ▲ Anaheim, CA 92801 ▲ (714) 635-4760

THE GREAT DEBATE

SPEAK 'N' SING 2 vs. SPEAK 'N' SING 1





WHATEVER YOUR CHOICE, YOU WIN
THE MOST VERSATILE SYNTHESIZERS ON ANY BUS

- Both the SPEAK 'N' SING 1 and the SPEAK 'N' SING 2 are ideally suited for a wide range of industrial, commercial, and entertainment applications such as: security systems, automatic telephone answering and originate devices, as an aid to the handicapped, teaching machines, and much more. Where an unlimited vocabulary is needed, the SPEAK 'N' SING 1 is ideal. When speech of exceptional quality is desired, the SPEAK 'N' SING 2 is perfect.
- Both synthesizers contain an on board FIFO memory buffer to reduce processor overhead and ease software development. None of the other popular bus oriented synthesizers has this feature.
- Each device has an on board audio power amplifier that easily drives an external speaker.
- Both synthesizers may be interrupt driven, however, the on board FIFO should reduce this need in most real time applications.
- A 60 page manual fully discusses speech, music, and sound effect synthesis.
- An 8 bit D/A converter is included on each board for the reproduction of MUSIC and SOUND EFFECTS.
- A single voice music interpreter is also included which allows one to develop MUSIC easily. For example, to synthesize a half note C sharp in the first octave, merely enter C#1H.
- Software is available on 5" or 8" disks in FLEX 2.0 or FLEX 9.0 formats. Included are many utility programs as well as MUSIC, SOUND EFFECTS, and SPEECH games.

SPEAK 'N' SING 1 SPEECH SYNTHESIS

The SPEAK 'N' SING 1 uses a special large scale integrated circuit, the SC-01 by VOTRAX, to reproduce any one of 64 phonemes at 4 inflections. These phonemes allow one to reproduce any word in English as well as many other languages. Typically only 5 or 6 phonemes are needed for each word. Supplied software includes a powerful speech editor with a preprogramed 600+ word vocabulary.

SPEAK 'N' SING 2 SPEECH SYNTHESIS

The SPEAK 'N' SING 2 speech is based on the National Semiconductor DIGITALKER synthesis technique. The high quality of the speech is incredible and must be heard to really be appreciated. Each word is reproduced by storing a single one byte code. The standard word set is listed below and space is provided on the board for additional vocabulary which may be purchased separately.

ONE	FOURTEEN	ARREST	PAUSE TONE	CENTER	HIGHER	MARK	PARENTHESIS	SET	A	N
TWO	FIFTEEN	ADDRESSED	20MS WAIT	EQUAL	HOUR	METER	PERCENT	SPACE	B	O
THREE	SIXTEEN	THOUSAND	40MS WAIT	ERROR	IN	MINE	PLEASE	SPEED	C	P
FOUR	SEVENTEEN	MILLION	80MS WAIT	FEET	INCHES	MILL	PLUS	STAR	D	Q
FIVE	EIGHTEEN	ZERO	160MS WAIT	FLOW	IS	MINUS	POINT	START	E	R
SIX	NINETEEN	ADAM	320MS WAIT	FUEL	IT	MINUTE	POUND	STOP	F	S
SEVEN	TWENTY	AMPERE	CENTI	GALLON	KILO	NEAR	PULSES	THAN	G	T
EIGHT	THIRTY	AND	CHECK	GO	LEFT	NUMBER	RATE	THE	H	U
NINE	FORTY	AT	COMMA	GRAM	LESS	OF	RS	TIME	I	V
TEN	FIFTY	CAMBEL	CONTROL	GREAT	LESSER	OFF	READY	TRY	J	W
ELEVEN	SIXTY	CASE	DARLER	GREATER	LIGHT	ON	RIGHT	UP	K	X
TWELVE	SEVENTY	CENT	DEGREE	HAVE	LOW	OUT	SS	VOLT	L	Y
THIRTEEN	EIGHTY	CORP	DOLLAR	HIGH	LOWER	OVER	SECOND	WEIGHT	M	Z

Write for information on additional word sets.

THE GOOD NEWS

SS-2 SPEAK 'N' SING 2 assembled, tested, manual, disk.	$239.95
WD-1 Two additional ROMs with 131 words ($44.95 when ordered with SS-2)	$49.95
SF-21 Additional software (Games, Sound Effects, Music)	$29.95
SF-22 More software (Games, Sound Effects, Music)	$29.95
SS-1 SPEAK 'N' SING 1 assembled, tested, manual, disk.	$219.95
SA-1 As above without SC-01 speech synthesizer chip (Allows Alford & Assoc. VS-1 owners to upgrade.)	$169.95
SF-1 Additional Software (Games, Sound Effects, Music)	$29.95
SF-2 More Software (Games, Sound Effects, Music)	$29.95

FOR THE MUSIC LOVER

This 4 voice Stereo Music Compiler easily allows the SPEAK 'N' SING 1, the SPEAK 'N' SING 2, or the NEWTECH Model 68 music synthesizer to reproduce 4 voice music. Written entirely in machine code, it compiles over 50 times faster than BASIC models. The compiler plays four voices simultaneously and allows tempo changes anywhere in the song. It also supports the reproduction of music in stereo. Other features include a 7 octave range, 45 different note durations, built in debugging capability, interface to FLEX files, IF statement for repeating sections and much more. Truely an incredibly powerful MUSIC compiler. ME-1 $39.95

1. BE SURE TO SPECIFY DISK SIZE AND FLEX FORMAT.
2. We accept MASTERCARD, VISA, COD, and CHECKS
3. Illinois residents add 5% sales tax.
4. Postage prepaid on US orders.
5. Overseas orders add $12 postage.
6. FLEX is a trademark of Technical Systems Consultants.

"WE'RE TALKING TO YOU"

Speech Systems

(312) 879-6880

38 W 255 DEERPATH ROAD
BATAVIA, IL 60510

CALL ANY DAY, ANYTIME FOR DEMO AND/OR TO ORDER. YOU MAY ALSO ORDER BY MAIL.

SMOOTH™ Software

Modem Program with Disk File Transfer (Instructions and Source Listing) $40.00
Super Modem Program (Manual with Instructions and Source Listing) 60.00
Specify Regular or Super, 6800 SSB, 6800 FLEX™, 6809 FLEX™, 5" or 8"

ALL IN ONE for only $45.00
Editor - Text Processor - Mailing Labels
Mailing Lists - Use any CRT terminal and printer

Supports Editing commands such as bottom, change, delete, find, insert (single line), input (multiple lines), list, next, overlay (with cursor editing, character deletion and insertion), o restrike (for selected darker text), print, restart, set, top, und rline, up, and verify.

Supports Text Processing commands such as block copy, block mov , centering, margin justification (widen and narrow), paging, and tabbing.

Mailing Lists and Labels. Use the same mailing list disk file (with protected areas) for both mailing labels and repeat letters. Repeat letters are personally addessed to each person or selected persons on the mailing list.

Most Powerful File Handler found in any editor. Append one file to the end of another, or insert (merge) one file into another as designated by the line pointer. Print specified lines to your printer or to a disk file. Edit files larger than the text buffer. Does not produce output files when not desired. D lete ask li s from the editor.

Printer commands. Control characters can be sent to the printer for format control either directly from the control terminal or by embedding them in the text. The set command contains interface initialization and character output routines to support the SWTPC MP-C interface as well as the standard serial and parallel interfaces. Jumpers are also provided to user supplied printer routines. User selects the port address (0 thru 7, A or B) thereby eliminating the need for the user to install printer software routin s. Editor can be initialized for either 4 or 16 addresses per port.

Editor allows exiting to either the monitor or DOS and then reenter (Warm Start) without destroying previously prepared text in the buffer. The Restart command erases contents in the buffer without the user having to reload the Editor.

The Editor allows the user to toggle between full duplex (no ec ho) and half duplex (echo) as needed. It responds to commands in both upper and lower case and can be used to create assembler source code and Basic programs as well as text.

Specify 6800 SSB, 6800 FLEX™ 6809 FLEX™, 5" or 8" 45.00
Printed source listing is available for an additional 35.00

Software by Technical Systems Consultants, Inc.

Flex™ (includes Editor and Ass m bler)	150.00
UniFLEX™ (includes one year maintenance and update)	450.00
Editor	50.00
Assembler	50.00
6809 Cross Assembler on 6800	100.00
68000 Cross Assembler on 6809	250.00
Text Processor	75.00
Basic	75.00
Extended Basic	100.00
Basic Precompiler (specify s andard or extended)	50.00
Multi-User Basic for S/09	150.00
Pascal (Flex™)	200.00
Pascal (UniFLEX™) (Add $75.00 for one year s maintenance an update)	225.00
Sort-Merge Package	75.00
6800 Flex™ Utilities	100.00
6809 Flex™ Utilities	75.00
Debug Package	75.00
Diagnostic Package	75.00

The following are available for 6800 only

Soap (stack oriented arithmetic processor)	25.00
Floating Point Package	25.00
Scientific Functions Package	25.00

Software by Microware Systems Corp.	YM&U	SOURCE	OBJECT
OS-9™ Level One Operating System	75.00		195.00
OS-9™ Level Two Operating System	75.00		500.00
BASIC09™	75.00		195.00
O -9™ Text Editor			125.00
OS-9™ Interactive Assembler			125.00
@ -9™ Interactive Debugger (Disk version)			50.00
CIS Cobol			895.00
Dynamite Disassembler			60.00
SUPER SLEUTH Disassembler System			99.00

SWTPC	Kit	Assembled
CT-64 Terminal kit (limited quantity)	295.00	295.00
DMF-2 8" DMA double density, double sided controller ard	N/A	595.00
MUB-68 Multi-User Board with Multi-User Basic	N/A	100.00
68/1 Computer	2 00	N/A
4K RAM Board	30.00	40.00
MP-S Serial Interface (single port)	40.00	60.00
MP-S2 Serial interface (dual port)	N/A	120.00
MP-LA Parallel interface (dual port)	40.00	60.00
MP-L2 Parallel interface (dual port)	N/A	120.00
MP-R Single voltage 2716 prom programmer	N/A	114.50
MP-N Calculator board	54.95	92.00
MP-T Interrupt timer	47.50	92.00
MP-8M 8K 4044 Memory board (limited quantity of kits)	160.00	275.00
S32 Universal Static Memory Board	N/A	124.50
MP-09 6809 CPU board	175.00	295.00
69 Chassis, P.S., 68B09 CPU, 8K, RAM, One Serial Port	660.00	799.00

6800 CPU and Disk Controller Boards

Due to the relative unavailability of these SWTPC items, please refer to the next column for ELEKTRA and GIMIX alternatives. Phone for SWTPC availability.

Universal 6XXX Bare Motherboard. SS-50/ 0C, 4/16 addresses per port 60.00
F & D Motherboard (Bare) 55.00

Connectors (10 pin, Titanium-Tin plated 5 microns for near gold quality)
Male with square cross section pins each .50
Female each .75

Microtime 6800 Cal ndar and Clock Board (assembled and tested) 105.00
Bare oard, connector, an documentation only of above 35.00
(See review Feb. 1980 '68' Micro Journal)
Microtime II 89.95
Data Mart 16K EPROM bareboard (2708 chips) 30.00

SMOOTH™ and **ELEKTRA™** are trademarks of AAA Chicago Computer Center
FLEX and **UniFLEX** are trademarks of Technical Systems Consultants, Inc.
OS-9 and **BASIC09** are trademarks of Microware Systems Corp.
GIMIX® and **GHOST®** are registered trademarks of GIMIX Inc.

D aler for GIMIX, SSB, SWTPC, Microware Systems Corp. and Technical Systems C nsultants, Inc.

Prices and Inventory are subject to change without advance notice.

ELEKTRA™ SS50 Computer Products

	Bare Board	Kit	Assembled
DPS Dual Port Serial Interface Board and Doc	20.00	60.00	80.00
DPP Dual Port Parallel Interface Board and Doc	20.00	60.00	80.00
Cable (Two required for each interface)			20.00
MB Motherboard and documentation	N/A	N/A	N/A
CPU-8/9 6808-6809 CPU (Run 6800 or 6809 software) (6808 is 6800 software compatible)	50.00	N/A	N/A
HUMBUG (from STAR-KITS) fo CPU-8/9 board			
2K version for 6800 (6808)			40.00
HUMBUG-09 for 6809			75.00

Other HUMBUG versions including video versions are available. (Specify system)

Cabinet (0.125" aluminum, 22" x 18-1/2" x 7", 2 cutouts for 5-1/4" disk drives, drive mount, line cord, line fuse, power switch, reset switch, abort switch, 70 cfm fan, EMI filter, 10 RS-232 cutouts) and power supply (15a 8v, 3a 16v, 3a -16v), use any current SS-50 or SS-50C motherboard	395.00
220v version of above for export	add 50.00
Cabinet without power supply	250.00
Power supply (15a 8v, 3a 16v, 3a -16v) Specify 110v or 220v	175.00
Power supply (20a 8v, 4a 16v, 4a-16v) Limited Quantity	125.00
5" disk regulator board with cables for 2 5-1/4" disk driv s	40.00
Filler plate for 5-1/4" cutout	10.00

5 1/4" DISK DRIVES with 30 day guarantee

Single head, single or double density capability, 40 tracks	$2 0.00
Single head, single or double density capability, 40 tracks, "flippy"	260.00
Double head, single or double density capability, 40 X 2 tracks	335.00
Single head, single or double density capability, 80 tracks	335.00
Double head, single or double density capability, 80 X 2 tracks	470.00
MPI - Service Manual	20.00
Dual drive cabinet for 5 1/4" drives with power supply, line cord, fuse, power switch, and power cables to driv s	125.00

Printers

Epson MX-80 (Centronics compatibl parallel interface)	495.00
(with Serial RS-232 interface option)	add 75.00
Spare print Head	39.95
Spare ribbon cartridge	15.00
Comet C. Itoh Comet I 1125 cps, 9 x 7, bidirectional, serial or parallel	475.00
Okidata Microline 82A 120 cps, 9 x 8, bidirectional, serial and parallel	495.00
Tractor for 82A	70.00
Optimal Technology, Inc. EP-2 A-79 Eprom Programmer	169.00
PM-0, PM-1, PM-5, PM-6, PM-7 Personality Modules	each 18.00
PM-2, PM-4	each 34.00
PM-3	26.00
PM-8	36.00
EP-2A-78-M-01 Eprom Programmer (User configures and supplies power)	79.95

GIMIX (The Ultimate)

6800 CPU Board		224.03
with timers		288.06
with baud rate option	add	30.00
with 2MHz option	add	15.00
2 MHz 6809 Plus CPU, time of day clock, battery backup, 1K NMOS RAM		578.05
CMOS RAM substitution		8.00
GIMIX Dynamic Address Translator		35.00
SWPTC compatible DAT		15.00
9511A Arithmetic Processor (4MHz)		312.00
9512 Arithmetic Processor (3MHz)		265.00
GMXBUG-09 (Terminal Based) 1K scratchpad required		98.65
Bootstrap Prom		30.00
Video Prom (includes bootstrap)		30.00
Manual and Source Listing only		38.62
Missing cycle detect card		38.23

Disk Controllers (All have data separators and can be us d with either single or double headed drives)

5" single density controller without 1771 chip	158.38
5" single density controller complete	198.48
5" and 8" single density controller complete	226.58
5" double density controller with variable precomp	348.28
DMA 5" AND 8" doubl density controller with variable precomp	588.68
GIMIX version of FLEX™ (without Editor and Assembler)	90.00
Double disk regulator card	68.22
Ribbon cable for two 5 1/4" disk drives (short)	34.96
Ribbon cable for two 5 1/4" disk drives (long)	39.96
Ribbon cable for two 8" disk drives (long)	44.26

Memory	CMOS WITH BAT. BACKUP	NMOS NO BAT. BACKUP
32K Static RAM Board with 16K of RAM installed		298.12
64K Static RAM Board with 24K of RAM installed	N/A	348.27
64K Static RAM Board with 32K of RAM installed	648.36	398.37
64K Static RAM Board with 48K of RAM installed	N/A	518.47
64K Static RAM Board with 56K of RAM installed	894.56	578.57
64K Static RAM Board with 64K of RAM installed	988. 4	638.67
16 Socket EPROM/ROM/RAM Board		238.32
8K Promboard (2708)		98.34
4K PPO 4K Prom Board and 2708 Prom Burner		198.00

I/O Boards

Single port 30 pin serial interface (Requires 1 cable set)	88.41
Dual port 30 pin serial interface (Requires 2 cable sets)	128.43
8 port 50 pin serial interface with baud rate generator	318.46
Dual port 30 pin parallel interface (Requires 2 cable sets)	88.42
8 port 50 pin parallel interface with interrupt generator	198.45
Cable sets for above boards (specify board)	22.95

Video Boards

64 or 32 X 16	198.71
80 X 24 without RAM character generator	398.74
80 X 24 with RAM character generator	458.76
High resolution (512 X 512 dot resolution)	996.77
2MHz 6809 PLUS Computer System with 56K Memory*	2498.28
Above System with #58 Controller and Special Software Pkg.*	2988.59
Above System with #68 Controller and Special Software Pkg.*	3248.49
*with CMOS RAM and Battery Backup	add 300.00
Mainframe (Chassis, PS, Switches, Fan, Motherboard, Baud Rate Gen.)	1198.19

Shipping and handling estimates:

Within the Continental U.S., please add $5.00 or 3% whichever is great r
Foreign, prepaid and add $10.00 or 10% whichever is greater (light items only). Heavy items must be prepaid and will be shipped Emery Air Freight Collect.
Ple se phone during consultation hours if questions arise regarding shipping fees.

Master Charge, Visa, and American Express honored

AAA Chicago Computer Center **See GIMIX Ad Pages 3 & 64**
120 Chestnut Lane, Wheeling, IL 60090
(312) 459-0450 hone Consultation available most weekdays fr m 4 PM to 8 PM

This ad is our catalog.

If you need M68000, Z8000, or 8086 Software, take it off our shelf.

Why wait up to 6 months for high quality 16-bit resident system software, when you can get it now?

Hemenway Associates can give you the software tools you need to get the most out of your 16-bit hardware. Because 8086, Z8000 and 68000 Systems Software is sitting on our shelves right now.

Single user operating systems, macro assemblers, text editors, floating point packages, multi-tasking operating systems, linking loaders and PASCAL compilers are all ready for immediate delivery.

You've heard about Hemenway Associates' system designs in the pages of EDN. But now you can have Hemenway-designed software for your 16-bit microprocessor. Our software is easily adaptable to any hardware configuration. From manufacturers evaluation boards to full-blown microcomputers. It's efficient, easy-to-use and backed by ongoing support.

So whether you're a microcomputer manufacturer, software developer or individual microcomputer owner, Hemenway has the software for you.

For the hard facts on Hemenway software, write or call: Hemenway Associates, 101 Tremont Street, Boston, Massachusetts 02108. Phone: 617-426-1931. TWX 710 321 1203. TELEX 921735.

HEMENWAY ASSOCIATES, INC.

When it comes to software, come to Hemenway.

DISK DRIVE WOES?
PRINTER INTERACTION?
MEMORY LOSS?
ERRATIC OPERATION?

Don't Blame The Software!

Power Line Spikes, Surges & Hash could be the culprit! Floppies, printers, memory & processor often interact! Our patented ISOLATORS eliminate equipment interaction AND curb damaging Power Line Spikes, Surges and Hash.

Pat. #4,259,705

- ISOLATOR (ISO-1) 3 filter isolated 3-prong sockets; integral Surge/Spike Suppression; 1875 W Maximum load, 1 KW load any socket $62.95
- ISOLATOR (ISO-2) 2 filter isolated 3-prong socket banks; (6 sockets total); integral Spike/Surge Suppression; 1875 W Max load, 1 KW either bank $62.95
- SUPER ISOLATOR (ISO-3), similar to ISO-1 except double filtering & Suppression $94.95
- ISOLATOR (ISO-4), similar to ISO-1 except unit has 6 individually filtered sockets $106.95
- SUPER ISOLATOR (ISO-11) similar to ISO-2 except double filtering & Suppression $94.95
- CIRCUIT BREAKER, any model (add-CB) ..... Add $ 8.00
- CKT BRKR/SWITCH/PILOT (-C S) ..... Add $16.00

AT YOUR DEALERS

Master-Card, Visa, American Express
Order Toll Free 1-800-225-4876
(except AK, HI, PR & Canada)

Electronic Specialists, Inc.
171 South Main Street, Natick, Mass. 01760
Technical & Non-800: 1-617-655-1532

MAGIC SPELL™

Now your 6800 or 6809 system can proof-read your text files for spelling and typographical errors in just minutes. MAGIC SPELL™ compares each word in your file against a master dictionary file, and displays or prints every word not found. The program is written in machine language and is extremely fast and compact — it will run in systems as small as 16K, and can proof-read documents much larger than can be held in memory.

MAGIC SPELL™ is supplied with a core dictionary containing thousands of common English words. The dictionary is easily modified by any text editor and can quickly be customized by the addition of technical terms or even names. Moreover, as new words are encountered in your texts, they may optionally be added to your dictionary as well. Hence the dictionary can easily be tailored to your own style of writing.

MAGIC SPELL™ is now available in versions for Technical Systems Consultants' MiniFlex, Flex 2, and Flex 9, as well as for Percom disk systems, and costs $89.29 complete with source code on disk. OS-9 and SSB disk versions will be available soon.

For further information write for our catalog or call (914) 241-0287. If you call in the late evening, turn on your modem and LIST MAGIC.DAT.

STAR-KITS
P.O. BOX 209
MT. KISCO, NEW YORK 10549

1080 IRIS DRIVE
CONYERS, GEORGIA 30207

SOUTHEASTERN MICRO SYSTEMS, INC.

404-922-1620

DECEMBER SPECIALS!

(Prices Good This Month Only)

DS-16
Serial Interface

- TWO ACIA PORTS
- TWO ON BOARD DB-25 CONNECTORS (on top of board)
- HAS ON BOARD BAUD RATE GEN
- BAUD RATE MAY BE PROGRAMMED FOR ON OR OFF BOARD USE
- BOARD CAN USE EXTERNAL BAUD RATE FROM BUSS OR RS-232
- BAUD RATE IS PROGRAMMABLE TO 32K BAUD
- FOR THE SS-30 BUSS
- ALL LINES BUFFERED
- ALL REGULATORS 1 amp. +5 +12 -12

BARE $19.95

DM-64
Dynamic Memory

- USES LOW COST 8116/4116 DYNAMIC RAM
- TOTAL TRANSPARENT REFRESH: OPERATES LIKE STATIC
- LOW POWER CONSUMPTION
- ADDRESSABLE IN 8-K BLOCKS
- CAN BE USED WITH 16K TO 64K
- EXTENDED ADDRESSING TO 16 BOARDS
- OPERATES ON SS-50 AND SS-50C BUSS
- SOLDER MASK AND SILKSCREENED BOARDS
- ALL MEMORY CHIPS ARE PRIME 8116-200ns

64K KIT $295.00
16K KIT $229.00

DDC-16
Disk Controller

- CAN BE PROGRAMMED FOR COMPATIBILITY WITH SWTP DC4
- WRITE PRECOMP/HARDWARE OR SOFTWARE SELECTABLE
- WRITE PROTECT HARDWARE OR SOFTWARE SELECTABLE
- SIDE SELECT SOFTWARE - SWTP/OTHER
- DS/DD
- ON BOARD DATA SEPARATOR (1691 & 4 phase clock)
- 4 or 16 ADDRESSED PER PORT
- WILL RUN SS/SD TO DS/DD

ASSEMBLED $250.00
KIT $200.00
BARE $50.00

ST-02
Video Board

- STAND ALONE VIDEO BOARD
- BAUD RATES 300-9600
- SCREEN PRINTER PORT
- STANDARD RS-232 INTERFACE
- 8 BIT KEY BOARD PORT
- PROGRAMMABLE SCREEN FORMAT

BARE $75.00

MODEM BOARD

- FOR SS-30 BUSS
- AUTO ANSWER
- AUTO DIAL
- PROGRAMMABLE BAUD RATE: 110/300 BAUD
- FILE TRANSFERS UNDER FLEX
- INCLUDES COMLINK PROGRAM FOR 6800 OR 6809 SOUTHWEST/FLEX SYSTEMS
- PROGRAM AVAILABLE ON 5¼" OR 8" DISKS

$100.00
BARE BOARD WITH PROGRAM ONLY

SM-56
Static Ram

- 2 MHZ OPERATION
- LOW POWER (2 amp)
- EXTENDED ADDRESSING
- ALL I/O LINES BUFFERED
- ADDRESSABLE IN 2K BOUNDARIES
- DESIGNED FOR 2K x 8 STATIC CHIPS

ASSEMBLED $550.00

4K KIT $149.00

US SHIPPING $3.50, FOREIGN ADD 10% (US FUNDS ONLY)

Master Charge, Visa, American Express Accepted

THOMAS INSTRUMENTATION

THE MACHINE TOOL, INDUSTRIAL SPECIALISTS IN BUSINESS ON A FULL TIME BASIS FOR 10 YEARS

NEW PRODUCT: **S-R/R**

48K 2MHz STATIC RAM/ROM CARD

- *24 2K blocks memory mapped on any 2K boundary
- *uses low power 2016P-2 (2128) RAM and/or 2716 ROM
- *mix 4K blocks of RAM and ROM
- *6800 and 6809 compatible
- *use on SS-50 and SS-50C buss
- *decoded for extended addressing
- *5 volts only
- *low power consumption (typ. ½ amp with 48K RAM)
- *gold connectors

Bare Board $49.00 2716 1MHZ $9.95 2016 P-2 2MHZ $16.50

A/T with 16K $250.00; with 32K $375.00; with 48K $495.00

A/T without memory chips $120.00

NEW ACCESSORIES FOR 68XX USERS:

SS-50/SS-50C EXTENDER CARD	$35.00
SS-30 EXTENDER CARD	$25.00
*Both cards assembled with a built in logic aid & gold edge connectors	
SS-30 WIRE-WRAP/PROTOTYPE BOARD (board only)	$20.00
*Pad spacing permits most standard sockets from 8 to 64 pins	
*Provision has been made for voltage regulators	

FEATURED PRODUCT: **SP-1 Bare card $49.00 Asm. + tested $195.00**

*A super prototype board *Card design includes

(3) 6821 6 parallel ports
(4) 6850 4 serial ports
(1) 6840 3 16 bit counter/timers
which are fully buffered and decoded

*Accomodates a mix of 38, 14 & 16 pin wire wrap sockets

*Pad spacing permits most standard sockets from 8 to 64 pins

MODEM CARD	B/C	$ 49.00
special parts kit		$195.00
A/T without extra features		$325.00
*SUPER CPU assembled with source listing without 2K EPROMS (2-2708)		$235.00
*Monitor in two 2708 EPROMS		$ 29.00
*CPU bare card, doc., & src.		$ 59.00
*VIDEO RAM asm. 7x9 chars 64x16		$195.00
*VIDEO RAM bare, doc, Xtal, src.		$ 49.00
*PARRALLEL I/O asm 100 I/O lines incl. 5 PIA's for 10 ports		$139.00
*PARALLEL I/O bare card & doc.		$ 49.00
*SS-50 WIRE-WRAP/PROTOTYPE bare		$ 39.00
*TRANSITION CARD asm.		$ 95.00
*TRANSITION CARD bare		$ 49.00

A/T with extra features	$395.00
Software obj. & src. on FLEX disk	$ 10.00
BACKPLANES AND MOTHERBOARDS	
*16 position SS-50	$80.00
*12 position SS-50	$60.00
* 8 position SS-50	$40.00
* 6 position SS-50	$30.00
* 4 position SS-50	$20.00
* 8 position SS-30	$39.00

**Connectors:
GOLD $1.60 ea. (M or F)
TIN M $.40 ea. F $.50 ea.

DEALERS FOR SWTPC, GIMIX, AND TSC

*All Thomas Instrumentation's cards come with full documentation including software source listings where applicable *All assembled cards are burned in at 150F and fully tested with Gold conn. *Bare card prices do not include edge connectors *See previous ads, write, or call for more detailed information.

THOMAS INSTRUMENTATION

168 EIGHTH STREET — AVALON, N.J. 08202 (609) 967-4280
NJ RES. INCLUDE 5% SALES TAX
CONT. USA INCLUDE $3.00 SHIPPING, CANADA $6.00, FOREIGN $12.00
MASTERCARD, VISA, and C.O.D. ACCEPTED

To satisfy in-depth questions regarding our products send $20.00 to receive full documentation, schematics, & source listings for all boards currently in production

JUDGE THE REST, THEN BUY THE BEST

Only GIMIX offers you **SOFTWARE SWITCHING** between **MICROWARE's OS-9** and **TSC's FLEX.** Plus you get the power of the GMXBUG system monitor with its advanced debugging utility, and memory manipulation routines. A wide variety of languages and other software is available for these two predominant 6809 Disk Operating Systems.

You can order a system to meet your needs, or select from the 6809 Systems featured below.

JUDGE THE FEATURES AND QUALITY OF GIMIX 6809 SYSTEMS

GIMIX' CLASSY CHASSIS™ is a heavyweight aluminum mainframe cabinet with back panel cutouts to conveniently connect your terminals, printers, drives, monitors, etc. A 3 position keyswitch lets you lock out the reset switch. The power supply features a ferro-resonant constant voltage transformer that supplies 8V at 30 amps, + 15V at 5 amps, and − 15V at 5 amps to insure against problems caused by adverse power input conditions. It supplies power for all the boards in a fully loaded system plus two 5 ¼" drives (yes! even a Winchester) that can be installed in the cabinet. The Mother board has fifteen 50 pin and eight 30 pin slots to give you the most room for expansion of any SS50 system available. 11 standard baud rates from 75 to 38.4K are provided and the I/O section has its own extended addressing to permit the maximum memory address space to be used. The 2 Mhz 6809 CPU card has both a time of day clock with battery back-up and a 6840 programmable timer. It also contains 1K RAM, 4 PROM/ROM/RAM sockets, and provides for an optional 9511A or 9512 Arithmetic Processor. The RAM boards use high speed, low power STATIC memory that is fully compatible with any DMA technique. STATIC RAM requires no refresh timing, no wait states or clock stretching, and allows fast, reliable operation. The system includes a 2 port RS232 serial interface and cables. All GIMIX boards use gold plated bus connectors and are fully socketed. GIMIX designs, manufactures, and tests in-house its complete line of products. All boards are twice tested, and burned in electrically to insure reliability and freedom from infant mortality of component parts. All systems are assembled and then retested as a system after being configured to your specific order.

56KB 2MHZ 6809 SYSTEMS WITH GMXBUX/FLEX/OS-9 SOFTWARE SELECTABLE

With #58 single density disk controller **$2988.59**
With #68 DMA double density disk controller **$3248.49**
to substitute Non-volatile CMOS RAM with battery back-up, add 300.00
for 50 Hz export power supply models, add 30.00

Either controller can be used with any combination of 5" and/or 8" drives, up to 4 drives total, have data recovery circuits (data separators), and are designed to fully meet the timing requirements of the controller I.C.s.

5 ¼" DRIVES INSTALLED IN THE ABOVE with all necessary cables

	SINGLE DENSITY		DOUBLE DENSITY		
	Formatted	Unformatted	Formatted	Unformatted	
40 track (48TPI) single sided	199,680	250,000	341,424	500,000	2 for **$709.00**
40 track (48TPI) double sided	399,360	500,000	718,848	1,000,000	2 for **900.00**
80 track (96TPI) single	404,480	500,000	728,064	1,000,000	2 for **900.00**
80 track (96TPI) double	808,960	1,000,000	1,456,128	2,000,000	2 for **1300.00**

Chart shows total capacity in Bytes for 2 drives.

Contact GIMIX for price and availability of 8" floppy disk drives and cabinets; and 5" and 8" Winchester hard disk system.

128KB 2Mhz 6809 DMA Systems for use with TSC's UNIFLEX or MICROWARES's OS-9 Level 2

(Software and drives not included) **$3798.39**
to substitute 128KB CMOS RAM with battery back-up, add 600.00
for each additional 64KB NMOS STATIC RAM board, add 639.67
for each additional 64KB CMOS STATIC RAM board, add 988.64
for 50 Hz export power supply, add 30.00

NOTE: UNIFLEX can not be used with 5" minifloppy drives.

GIMIX has a wide variety of RAM, ROM, Serial and Parallel I/O, Video, Graphics, and other SS50 bus cards that can be added now or in the future. Phone or write for more complete information and brochure.

THE SUN NEVER SETS ON GIMIX USERS

GIMIX Systems are found on every continent, except Antarctica. (Any users there? If so, please contact GIMIX so we can change this.) A representative group of GIMIX users includes: **Government Research and Scientific Organizations** in Australia, Canada, U.K., and in the U.S.: NASA, Oak Ridge, White Plains, Fermilab, Argonne, Scripps, Sloan Kettering, Los Alamos National Labs, AURA. **Universities:** Carleton, Waterloo, Royal Military College, in Canada; Trier in Germany; and in the U.S.: Stanford, SUNY, Harvard, UCSD, Mississippi, Georgia Tech. **Industrial users** in Hong Kong, Malaysia, South Africa, Germany, Sweden, and in the U.S.: GTE, Becton Dickinson, American Hoechst, Monsanto, Allied, Honeywell, Perkin Elmer, Johnson Controls, Associated Press, Aydin, Newkirk Electric, Revere Sugar, HI-G/AMS Controls, Chevron. **Computer mainframe and peripheral manufacturers,** IBM, OKI, Computer Peripherals Inc., Qume, Floating Point Systems. **Software houses;** Microware, T.S.C., Lucidata, Norpak, Talbot, Stylo Systems, AAA, HHH, Frank Hogg Labs, Epstein Associates, Softwest, Dynasoft, Research Resources U.K., Microworks, Analog Systems, Computerized Business Systems.

TO ORDER BY MAIL

SEND CHECK OR MONEY ORDER OR USE YOUR VISA OR MASTER CHARGE. Please allow 3 weeks for personal checks to clear.

U.S. orders add $5 handling if order is under $200.00. Foreign orders add $10 handling if order is under $200.00.

Foreign orders over $200.00 will be shipped via Emery Air Freight C.I.F./D.C.T., and we will charge no handling. All orders must be prepaid in U.S. funds. Please note that foreign checks have been taking about 8 weeks for collection so we would advise wiring money, or checks drawn on a bank account in the U.S. Our bank is the Continental Illinois National Bank of Chicago, account #73-32033. Visa or Master Charge also accepted.

GIMIX INC. reserves the right to change pricing and product specifications at any time without further notice.

GIMIX Systems are chosen by the Pros because of quality, reliability and features.

GIMIX INC.

The Company that delivers

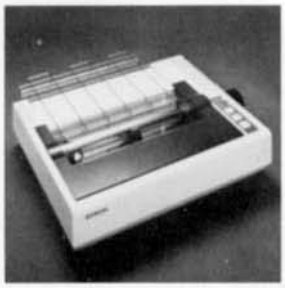
Quality Electronic products since 1975

1337 WEST 37th PLACE, CHICAGO, IL 60609

(312) 927-5510 • TWX 910-221-4055

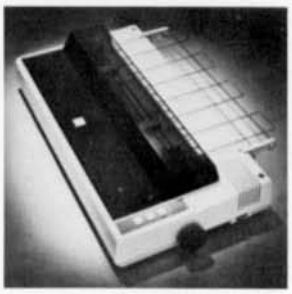
EPSON DOT MATRIX PRINTERS

The Epson MX-80



80-Column
$495.00

The Epson MX-100



136-Column
$798.00

MX-70 $375.00 MX-80FT $625.00

STARWRITER DAISY WHEEL PRINTERS

Model FP1500-25 $1775.00
25CPS

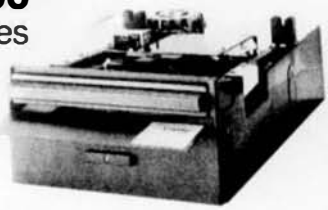
Model FP1500-40 $1995.00
40CPS

FACTORY FRESH — LATEST PRODUCTION — BRAND NEW

2 for $1059.50
Add: $7.50 each - Shipping and Handling
Offers quality and features found in drives costing much more! ■ Single or Double Density ■ Double-sided Drive ■ Door Lock INCLUDED ■ Write-Protect INCLUDED ■ 90 Day Warranty ■ Compatible with Shugart 850 ■ Low Power Operation ensures LONGER LIFE!!

Remex RFD4000
8" Floppy Disc Drives
Double sided . . .
Double density! !

$559.95



DUAL 8" DISK DRIVE CABINET WITH POWER SUPPLY AND CABLE
$325.00

REMEX TECHNICAL MANUAL $9.95

Cannot guarantee supply will last if paid for by personal check as personal checks require 3-4 weeks to clear - Recommend Money Order or Certified Bank Check.

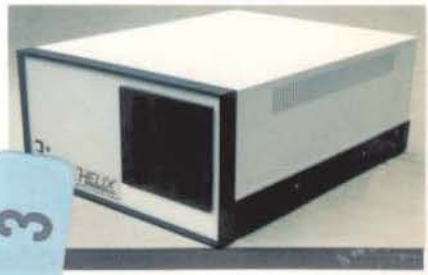
SOUTH EAST MEDIA
P.O. Box 794 Chattanooga TN 37443
1-615-842-4601

WE SELL
ONLY DATALIFE
DISKS

68 Micro Journal
5900 Cassandra Smith Rd.
Hixson, TN 37343

Second Class Postage Paid
At Chattanooga, TN
ISSN 0194-5025

HELIX™



THE MAINFRAME

- Industry Standard Optima™ Cabinet
- Largest Constant Voltage Power Supply in the industry
- S-64 Bus gives 16 Bit Power and S-50 Bus Compatibility
- 10 Main (S-64) Slots
- 14 I/O (S-30) Slots plus 2 On-board
- On-board Baud Rate Generator to 38.4Kb
- Space and Power for two 5¼" Disk Drives
- Full Address Decoding for I/O Slots
- Two RS-232 Serial and Two parallel Ports On-board
- Single Board Construction for Reliability
- Faraday Shielded Bus Lines give "Text Book Clean" Signals

THE PROCESSORS

6809

- Standard 2 MHz Operation
- Standard DAT Compatible with GIMIX and SWTPC
- Standard 6840 Interval Timer
- Standard 1K Scratchpad RAM
- Standard Clock/Calendar with Battery
- Provision for Programmers Console

68000

- Standard 8 MHz Operation
- Memory Management Hardware
- Provision for Programmers Console
- 16 Bit Power and 8 Bit Compatibility

The HELIX™ computer system represents the latest advance in S-50 bus computer systems. Relying on the physical nature of S-50 bus connectors to guarantee compatibility, the HELIX adds 14 bus lines (becoming S-64) to allow a 68000 processor to operate with full 16 bit data transfer and 24 bit addressing, while at the same time providing full interchangability with existing S-50 components.

Offered with a selection of processors, memories, and peripheral controllers, a HELIX system can be configured for applications ranging from advanced hobbyist to multiterminal time-sharing.

Designed to offer the utmost in speed, reliability, and utility at a reasonable price, it represents a new standard of quality for those who require a professionally designed computer for professional use.

THE POWER SUPPLY

- Ferro-resonant Transformer for Line Noise and Under-Voltage Protection
- Conservative 25 Amps at 8.5 Volts
- Conservative 5 Amps at ±18 Volts
- Conservative Component Rating for Reliability

THE COMPONENTS

- Fully Socketed
- Gold Plated Bus Connectors
- Only "B" Series 68XX Components Used
- Only Top Grade Logic Circuits Used
- Industrial Grade Components Throughout

THE MEMORIES

DM-64

- Field Proven
- Proprietary Memory Control Logic
- Fully Transparent Refresh
- Tested at 2.5 MHz Operation

DM-512

- 512K Bytes on a Single S-64 Board
- 16 Bit Power and 8 Bit Compatibility
- Runs in Existing S-50 Systems where Physical Space Allows
- Full 24 Bit Addressing
- Fully Transparent Refresh

THE PRICES

Because of the variety of configurations possible, full pricing cannot be given. Representative prices are:

- 64K 6809 HELIX $1995
- 64K 68000 HELIX $2595
- 512K 6809 HELIX $4450
- 512K 68000 HELIX $4995

HAZELWOOD COMPUTER SYSTEMS

7413 N. Lindbergh, Hazelwood, Missouri 63042 (314) 837-3466

Dealer and OEM Inquiries Invited. We support our Dealers.

Optima is a Trademark of Scientific-Atlanta